Floppy

- ★ Commodore-Floppys im Vergleich
- ★ Großer Vergleichstest: Das leisten Speeder für die 1541
- ★ Der komplette Floppy-Kurs

Datei-verwaltung für Einsteiger und Profis

- ★ Ausführliche Grundlagen in Basic und Assembler
- ★ So arbeitet man mit dBase II

Viele Listings für Floppy und Datasette

- ★ schnelle Kopierprogramme
- ★ Diskettenverwaltung für C 64 und C 128
- ★ Spitzen Diskettenmonitore
- ★ Neues 64'er-DOS: Jetzt noch besser

Alle Programme auch auf Diskette erhältlich

64ER ONLINE

Die Daten sicher im Griff

Das herausragende Merkmal eines Computers mit angeschlossenem Massenspeicher ist seine Fähigkeit, nahezu unbegrenzte Mengen von Daten zu speichern, zu sortieren, zu verwalten und ansprechend auszudrucken. Um welche Art von Daten es sich dabei handelt, ist dem Computer vollkommen egal. Ob es sich um eine Verwaltung der Directory-Einträge oder die Organisation der Schallplattensammlung handelt, der C64 erledigt die ihm gestellten Aufgaben schnell und sorgfältig, sofern entsprechende Programme im Computer und externe Massenspeicher dazu vorhanden sind.

Welche Diskettenlaufwerke für die Systeme C16, C64, Plus/4 und C128 am besten geeignet sind, erfahren Sie in einem großen Vergleich der Floppy-Stationen 1541, 1551, 1570 und 1571. Ein wichtiges Kriterium ist sicherlich, welches Laufwerk für welchen Computer ideal oder nur mit gewissen Einschränkungen der Kompatibilität geeignet ist. Da die Laufwerke von Commodore allesamt nicht zu der schnellen Truppe zählen, stellen wir Ihnen in einem ausführlichen Bericht die aktuellsten Floppy-Beschleuniger vor. Nach Studium dieses Artikels kennen Sie die Vor- und Nachteile der bekannnten Speeder und können die richtige Kaufentscheidung für Ihren speziellen Anwendungsfall treffen.

Für die Besitzer der 1541-Floppy haben wir den vollständigen Floppy-Kurs der Ausgaben 10/84 bis 6/85 abgedruckt. Hier plaudert der Autor der beiden Bücher »Die Floppy 1541« und »Die Floppy 1570/1571«, Karsten Schramm, aus seiner Trickkiste.

Weitere Kurse befassen sich mit der Programmierung von selbstgeschriebenen Datenverwaltungsprogrammen sowohl in Basic als auch in Assembler. Hier teilt der bekannte Buchautor Said Baloui sein umfangreiches Wissen in leicht verständlicher Form mit. Dem Renner unter den professionellen Datenverwaltungsprogrammen - dBase II - ist ein Einführungskurs gewidmet.

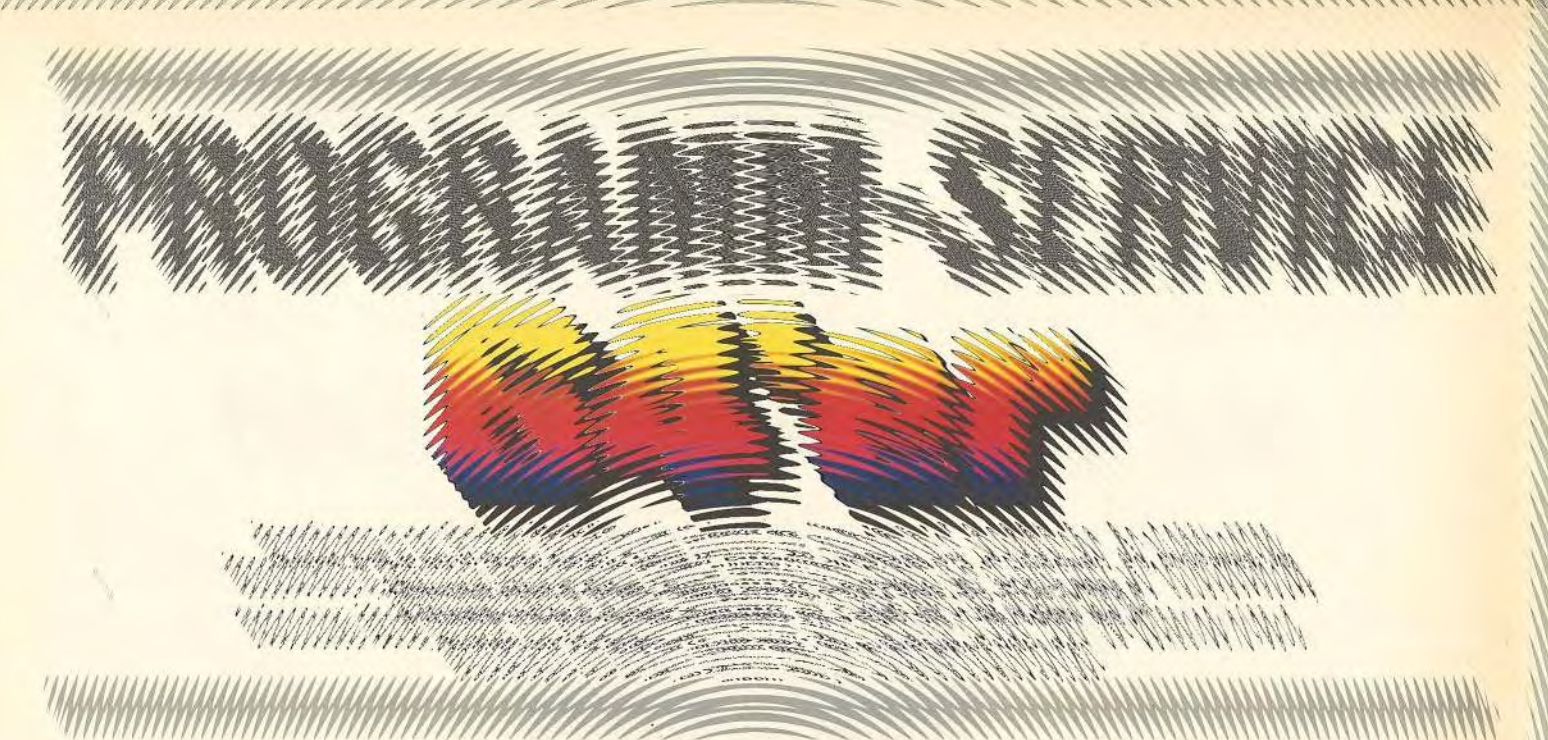
Nicht nur theoretisches Wissen wird in diesem Sonderheft vermittelt, auch die Praktiker kommen auf ihre Kosten. So gibt es zahlreiche Programme der verschiedensten Kategorien zum Abtippen. Wer keine Zeit beim Kopieren von Disketten verschwenden will, bekommt ein Kopierprogramm, mit dem in 90 Sekunden (zuzüglich Diskettenwechsel) eine komplette Diskette dupliziert ist. Auch das Formatieren der flexiblen Datenträger ist nunmehr in nur 10 Sekunden, ohne Verzicht auf die Qualität, zu erledigen. Mit einen raffiniertem Programm werden die bisher ungenutzten Spuren 36 bis 40 für die Speicherung zusätzlicher Daten zugänglich gemacht. Wem die normale Darstellung des Inhaltsverzeichnisses (Directory) einer Diskette nicht gefällt, der erhält einige Programme, mit denen der Ausdruck optimiert und der Informationsgehalt gesteigert werden kann.

Daß Gutes noch besser werden kann, beweisen wir mit dem neuen 64'er-DOS V4. Diese neue Version des 64'er-DOS aus der Ausgabe 3/86 beinhaltet den kompletten Monitor SMON und viele zusätzliche, komfortable Monitor- und Basic-Befehle. Es wird so zu einem vorzüglichen Hilfsmittel für Programmierer.

Zahlreiche weitere Programme runden dieses Sonderheft ab. Für den Datenschutz werden bei einem Löschvorgang nicht nur die Einträge im Inhaltsverzeichnis eliminiert, sondern die Daten tatsächlich »physikalisch« von der Diskette entfernt. Anhänger schneller Sortierroutinen dürfen sich auf Quicksort mit einem neuen Algorithmus freuen. 1000 Elemente sind in nur drei Sekunden geordnet. C128-Besitzer erhalten zum eingebauten Monitor zusätzlich einen Diskettenmonitor und ein universelles Autostartprogramm.

Alles in allem wieder ein Sonderheft, aus dem sowohl der Anfänger in Sachen Datenverwaltung und Massenspeicher, als auch der Profi wertvolle Tips und Informationen ziehen kann.

(Albert Absmeier)



INHALT

Vorwort

Hardware-Test

Grundlagen

Bücher

Tips und Tricks

Die Speicherriesen

Im folgenden Artikel nehmen wir für Sie die wichtigsten Diskettenlaufwerke von Commodore einmal unter die Lupe. Wir geben Ihnen Tips zum Kauf und zeigen Gemeinsamkeiten und Unterschiede aller Geräte. Dabei erfahren Sie auch, welches Gerät zu welchem Computer paßt.

Spätestens wenn Sie mit Ihrem Computer nicht mehr nur noch spielen, sondern auch ernsthafter arbeiten wollen, reicht ein Kassettenrecorder als Massenspeicher nicht mehr aus. Sie benötigen zumindest ein Diskettenlaufwerk, das sowohl in bezug auf die Geschwindigkeit als auch in bezug auf die Datensicherheit der Datasette einiges voraus hat.

Nun kommt jedoch die Entscheidung. Welches Laufwerk paßt am besten zu Ihrem Computer? Wieviel Geld wollen Sie investieren?

Diese beiden Punkte zeigen, wie schwer einem die Entscheidung fallen kann, zumal die Diskettenstationen von Commodore ansonsten sehr viele Gemeinsamkeiten aufweisen.

Mehr Gemeinsamkeiten als Unterschiede

Wir wollen Sie nun in Ihrer Entscheidungskraft ein wenig stärken, indem wir Ihnen alle wichtigen Diskettenlaufwerke von Commodore für den C 64, C 128, C 16 und Plus/4 einmal vorstellen. Wenn Sie bereits Besitzer eines Diskettenlaufwerks sind, dann wird Ihnen der Artikel helfen, Programme an andere Laufwerke anzupassen oder die Eigenheiten Ihres speziellen Laufwerks auszunutzen.

Die vier Geräte, mit denen wir uns beschäftigen, tragen die Bezeichnungen 1541, 1551, 1570 und 1571 und wurden insbesondere für die Computer C 64 (1541), C 16 und Plus/4 (1551) und C 128 (1570, 1571) entwickelt. Das schließt natürlich nicht von vornherein aus, daß die Laufwerke nicht auch unter den Computern ausgetauscht werden können.

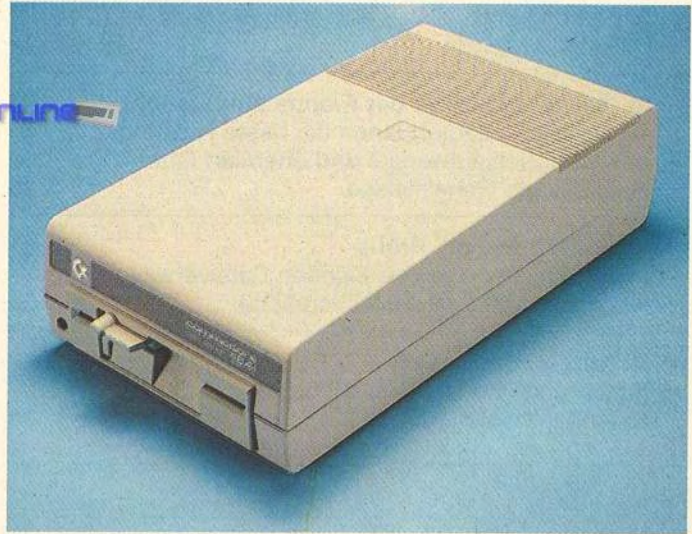


Bild 1. Die 1541 (neue (oben) und alte Version)

Das Konzept des Artikels erlaubt es Ihnen, anhand der Tabellen einen schnellen und vergleichenden Überblick zu bekommen. Im Text werden wir jedes Gerät einzeln behandeln und dessen Eigenschaften herausstellen, um Ihnen den Anwendungs- und Einsatzbereich etwas vor Augen zu führen.

Der Stammvater

An erster Stelle unserer Ausführungen steht natürlich die 1541 (Bild 1). Sie wurde sehr schnell als direkte Nachfolgerin der 1540 auf den Markt gebracht, um auch für den C 64 die Welt der Massenspeicher zu erschließen. Die 1540 war ursprünglich nämlich nur für den Anschluß an einen VC 20 vorgesehen und konnte am C 64 nicht betrieben werden.

Zuerst beschäftigen wir uns ein wenig mit der Hardware. Wenn die 1541 aufgeschraubt wird, dann kommt sofort die Platine mit den elektronischen Bauelementen zum Vorschein (Bild 2). Sie enthält neben dem 6502-Mikroprozessor zwei 6522-Interface-Bausteine (VIA), 2 KByte statisches RAM, 16 KByte ROM und einen »Diskcontroller« in Form eines Logic-Arrays. Fast alle übrigen Bauteile bilden den Analogteil der Platine und sind für die Schreib- und Lesesteuerung der 1541 zuständig.

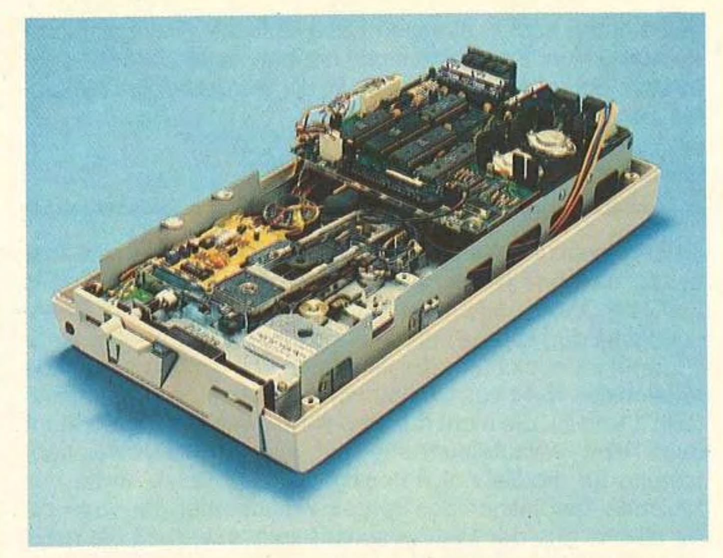
Es kann durchaus sein, daß sich in Ihrem Laufwerk eine etwas andere Platine als auf unserem Foto (Bild 2) befindet. Insgesamt gibt es mindestens vier verschiedene Versionen der 1541, wobei im Betriebssystem jedoch keine grundlegenden Änderungen vorgenommen wurden.

Wenn Sie Tabelle 1 betrachten, dann sehen Sie dort die im einzelnen aufgeführten technischen Daten der 1541. Es handelt sich hier um ein Prozessorsystem mit 1 MHz Taktfrequenz. Die Verbindung zum Computer erfolgt durch den seriellen Bus, der sich als Standard bei Commodore eingebürgert hat. Dieser Bus läßt Übertragungsraten bis zu 1200 bit/s pro Sekunde zu, und die 1541 liegt in der Geschwindigkeit damit ganz hinten unter allen Diskettenstationen von Commodore.

Für die 1541 gibt es als Abhilfe zur Zeit jedoch sehr viele Systeme zur Beschleunigung, die oft Geschwindigkeiten erreichen, die um den Faktor 30 über der Originalgeschwindigkeit liegen. Diese Beschleuniger nutzen die Tatsache, daß nicht das Laufwerk an sich so langsam ist, sondern lediglich die Schnittstelle zum Computer. Sie finden in dieser Ausgabe einen großen Vergleichstest, für die bekanntesten Beschleunigungssysteme, der sämtliche Stärken und Schwächen dieser Produkte aufzeigt.

Speicherkapazität	174 848 Byte total 169 984 Byte frei
Directoryeinträge	144
Anzahl der Spuren	35
Anzahl der Sektoren (Blocks)	683 insgesamt 664 frei 17 bis 21 pro Spur
Kapazität eines Sektors	256 Byte (mit Linkpointern)
Aufzeichnungsformat	GCR (...)
Aufzeichnungsrate	250 000 bit/s Abschnitt 1 266 664 bit/s Abschnitt 2 285 712 bit/s Abschnitt 3 307 688 bit/s Abschnitt 4
Bestückung	6502 Mikroprozessor VIA 6522 (2) 2 KByte statisches RAM Logic-Array als Diskcontroller
Taktfrequenz	1 MHz
Übertragungsrate	1200 bit/s seriell

Tabelle 1. Die technischen Daten der 1541



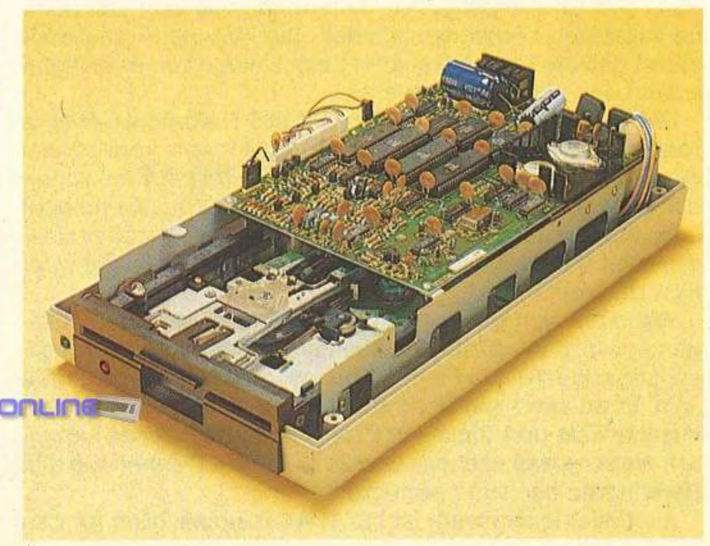


Bild 2. So sieht die 1541 von innen aus

Bei der 1541 gibt es auf dem Markt zur Zeit zwei verschiedene Ausführungen von Laufwerken. Dabei handelt es sich einmal um ein Laufwerk mit Klappverschluß und zum anderen um ein Laufwerk mit Knebelverschluß. Haben Sie die Wahl zwischen beiden Laufwerkstypen, so ist das mit dem Knebelverschluß vorzuziehen, da es eine bessere Mechanik aufweist. Der Knebelverschluß hat seinen Namen von der Art der Diskettenfixierung im Laufwerk. Hier wird ein Hebel senkrecht nach unten geklappt und das Laufwerk damit verschlossen. Beim Klappverschluß drücken Sie eine Klappe nach unten, und die Diskette wird verriegelt.

Die Mechanik der 1541 ist ziemlich starken Belastungen ausgesetzt. Da ist einmal die Temperatur. Sie wird durch die beiden integrierten Festspannungsregler auf der Platine stark erhöht und sorgt damit unter Umständen für ein Verstellen des Laufwerks unter extremen Einsatz. Zum anderen unterstützt auch das Betriebssystem nicht gerade eine Schonung der Mechanik. Bei Lesefehlern und beim Formatieren von Disketten wird der Schreib-/Lesekopf der 1541 auf den Nullanschlag zurückgefahren, was sich durch ein ziemlich häßliches Rattern bemerkbar macht.

Dieses Anschlagen ist im Zusammenhang mit hoher Betriebstemperatur der Hauptgrund für verstellte Laufwerke, weshalb die 1541 gerade im Dauerbetrieb schonend behan-

delt werden sollte; also zum Beispiel keine »Formatierstunden« einlegen. Es sollten immer höchstens 10 Disketten hintereinander formatiert werden, um ein Verstellen des Tonkopfes zu vermeiden.

Die neue 1541

Im Gegensatz zu allen anderen Laufwerken von Commodore besitzt die 1541 keine Anlaufsteuerung beim Einlegen einer Diskette. Diese Anlaufsteuerung schaltet dabei in der Regel kurzzeitig den Laufwerksmotor ein, um das Zentrieren einer Diskette zu erleichtern. Es ist jedoch seit kurzem eine neue Version der 1541 von Commodore auf den Markt gekommen (Bild 1 und 2), die nicht nur eine völlig neue Platine, sondern auch eine Anlaufsteuerung und eine Lichtschrankenabtastung für die Spur-Null-Position besitzt. Damit dürfte das typische Anschlagen des Schreib-/Lesekopfes der 1541 ein für allemal aus der Welt geschafft sein. Inwieweit die neue 1541 jedoch kompatibel zur »alten« ist, muß sich erst noch herausstellen. Erkennen können Sie das neue Laufwerk daran, daß der Laufwerksmotor beim Einlegen einer Diskette anläuft, um diese zu zentrieren.

Der Befehlssatz der 1541 ist für alle Diskettenlaufwerke von Commodore die Grundlage. Er erlaubt sehr komfortabel das Manipulieren von Disketten und wichtigen Funktionen der 1541. Das fängt beim Erstellen von Dateien an und geht über das Aufbringen eines Kopierschutzes bis hin zu professionellen Nibble-Kopierprogrammen wie Turbo Nibbler oder Fast Hack'em.

Wenn Sie sich genauer mit diesen Einzelheiten beschäftigen wollen, dann dürfte der, ebenfalls in dieser Ausgabe, vollständig abgedruckte Floppy-Kurs genau das Richtige für Sie sein. Er erläutert sämtliche Befehle der 1541 und gibt interessante Tips und Tricks zur Programmierung dieses sicherlich »außergewöhnlichen« Geräts. In Tabelle 2 sehen Sie den Befehlssatz der 1541 abgedruckt.

Als Diskettenlaufwerk ist die 1541 hauptsächlich für den

Befehl	Wirkung
NEW	Formatieren einer Diskette
INITIALIZE	Initialisieren einer Diskette
VALIDATE	Wiederherstellen der BAM (...)
COPY	Kopieren von Files
RENAME	Umbenennen eines Files
SCRATCH	Löschen eines Files
POSITION	Setzen des Recordzeigers
B-R	Lesen eines Sektors
B-W	Schreiben eines Sektors
B-A	Belegen eines Sektors in der BAM
B-F	Freigeben eines Sektors in der BAM
B-E	Ausführen eines Programms in einem Sektor
B-P	Setzen des Pufferzeigers in einem Sektor
M-R	Lesen von Speicherinhalten des Laufwerks
M-W	Schreiben von Speicherinhalten
M-E	Starten eines Maschinenprogramms im Laufwerk
U1	Lesen eines Sektors
U2	Schreiben eines Sektors
U3 bis U8	Programmstarts im Benutzerpuffer
U:	Reset ausführen
&	Laden und Starten eines Maschinenprogramms

Tabelle 2. Der Befehlssatz der 1541

Speicherkapazität	174 848 Byte total 169 984 Byte frei
Directoryeinträge	144
Anzahl der Spuren	35
Anzahl der Sektoren (Blocks)	683 insgesamt 664 frei 17 bis 21 pro Spur
Kapazität eines Sektors	256 Byte (mit Linkpointern)
Aufzeichnungsformat	GCR (...)
Aufzeichnungsrate	250 000 bit/s Abschnitt 1 266 664 bit/s Abschnitt 2 285 712 bit/s Abschnitt 3 307 688 bit/s Abschnitt 4
Bestückung	6510T Mikroprozessor TIA 6525 6523A 2 KByte statisches RAM Logic-Array als Diskcontroller
Taktfrequenz	1 MHz
Übertragungsrate	zirka 6000 bit/s parallel

Tabelle 3. Die technischen Daten der 1551

C 64 entwickelt worden. Sie läuft aber auch am C 16, C 116 und am Plus/4. Für den Commodore 128 kann die 1541 ebenfalls verwendet werden. Hier gibt es jedoch in der Regel Probleme mit der Geschwindigkeit, doch dazu später mehr.

Daß die 1541 am C 16 und am Plus/4 betrieben werden kann, heißt noch lange nicht, daß sie auch für diese Computer entwickelt wurde oder umgekehrt. Äußerlich wird das schon dadurch deutlich, daß hier ein ziemlich unschöner Farbkontrast auftritt. Der C 16/C 116 und der Plus/4 sind schwarz, die 1541 hingegen erstrahlt in einem hellen Beige.

Neben der 1541 für den C64 gibt es für C16- und Plus/4-Besitzer die 1551, die von Commodore speziell für den C16 und den Plus/4 entwickelt wurde.

Äußerlich präsentiert sie sich in einem eleganten schwarzen Gehäuse mit einem Knebelverschluß. Von allen Laufwerken von Commodore handelt es sich bei der 1551 sicherlich um den größten »Exoten«.

Laufwerk mit Parallelkabel

Im Gegensatz zu allen anderen Einzellaufwerken (die Laufwerke mit IEEE-488-Bus sind hier ausgenommen) arbeitet die 1551 mit einer Parallelverbindung zum Computer. Das bedeutet natürlich, daß kein Anschluß an die Buchsen des seriellen Bus mehr erfolgen kann. Die 1551 tritt vielmehr über den Expansion-Port mit dem Computer in Kontakt.

Dieser Anschluß hat zwei Konsequenzen:

Erstens braucht der Anwender einen Anschluß der preiswerten 1551 an den C 64 erst gar nicht in Betracht zu ziehen, da die Belegung des Expansion-Port am C 64 nicht mit der des C 16 und Plus/4 übereinstimmt.

Und zweitens kann man davon ausgehen, daß die 1551 am C 16 und Plus/4 schneller arbeitet als die 1541. In der Tat ist die Verarbeitungsgeschwindigkeit hier etwa viermal höher, was durch die höhere Übertragungsrate einer Parallelverbindung hervorgerufen wird.

Neben der Parallelverbindung wurden bei der 1551 noch ein paar andere Dinge gegenüber der 1541 verbessert. So verfügt die 1551 zum Beispiel über eine Anlaufsteuerung, die den Laufwerksmotor kurz anfahren läßt, wenn man eine Diskette einlegt.

Schraubt man die 1551 auf (Bild 3), fällt einem sofort die neue Platine ins Auge, die gegenüber der 1541 gravierende Unterschiede aufweist. Da ist einmal der Mikroprozessor.

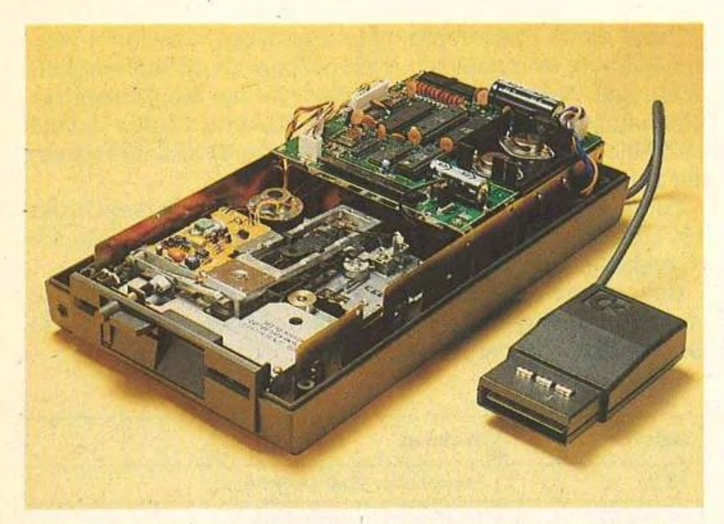


Bild 3. Die 1551 präsentiert sich mit einer äußerst kleinen Platine. Deutlich ist das Parallelkabel zu sehen.

Hier handelt es sich im Gegensatz zur 1541 (mit Prozessor 6502) um einen 6510T. Der 6510 findet auch im C 64 Verwendung, da er einen eingebauten I/O-Port besitzt und so seine Speicherverwaltung steuern kann.

Anstatt zweier VIA 6522 wie in der 1541 findet man in der 1551 einen einzigen TIA 6525. TIA steht für »Triport Interface Adapter«. Wir haben es also mit einem Interface-Baustein zu tun, der über drei 8-Bit-Schnittstellen verfügt. Das erlaubt einen vollständigen Ersatz der zwei VIA 6522. Der TIA 6525 findet übrigens auch schon in den Personal Computern der Reihe CBM 700 Verwendung. Es handelt sich hierbei also keineswegs um eine Neuentwicklung.

Überraschend an der 1551 ist die Platine, die mindestens um die Hälfte kleiner als die der 1541 ist. Diese Miniaturisierung wird durch ein neuartiges Bauteil ermöglicht, das entfernt an einen Hybridschaltkreis erinnert, jedoch den gesamten Analogteil der Diskettenstation enthält. Mit Analogteil ist dabei die Schreib-/Leseelektronik mit den Verstärkerstufen gemeint.

Die neue Version der 1541 enthält übrigens ebenfalls diesen integrierten Baustein, so daß auch hier eine sehr viel kleinere Platine entwickelt werden konnte.

Bei der 1551 handelt es sich, wie schon bei der 1541, um ein Prozessorsystem mit 1 MHz Taktfrequenz. Die technischen Daten entnehmen Sie bitte der Tabelle 3.

Der Befehlssatz der 1551 ist fast identisch zu dem der 1541. Bei Tests konnte kein Programm gefunden werden, das mit einem C 16 und einer 1541 läuft und mit einer 1551 nicht. Die Kompatibilität ist also offensichtlich sehr hoch, wobei die 1551 jedoch im Gegensatz zur 1541 zusätzlich zwei sehr interessante Befehle enthält. Einmal kann die Anzahl der Leseversuche bei Fehlern und zweitens der Sektorabstand beim Schreiben von Dateien ganz bequem eingestellt werden. Die Befehlsliste ist in Tabelle 4 abgedruckt.

Das »Zwischending«

Beim C 64 konnte man sich noch mit Notlösungen behelfen, wenn es um schnelles Laden ging. Da der C 128 von Commodore jedoch als Personal Computer verkauft wird, reichte die 1541 nicht mehr aus.

Hier ging es jedoch nicht unbedingt um die Geschwindigkeit des Diskettenlaufwerks. Ausschlaggebend für die Entwicklung neuer Laufwerke war vielmehr die Eigenschaft des C 128, auch CP/M-fähig zu sein.

Die 1541 und auch die 1551 ist technisch nicht in der Lage, Disketten zu lesen oder zu beschreiben, die im MFM-Format (Diskettenformat vieler anderer Laufwerke) geschrieben worden sind. Um das zu realisieren, mußte ein Diskettenlaufwerk her, das einen MFM-Diskcontroller enthält. Hier verwendet man üblicherweise den WD 1770 von Western Digital, der in sehr vielen Laufwerken enthalten ist.

Auch die 1570 gehört zu den Laufwerken, die diesen Diskcontroller beinhalten. Die Mechanik der 1570 entspricht weitgehend der Mechanik der 1541. Lediglich die Elektronik wurde komplett überarbeitet.

Schraubt man die 1570 auf, so erscheint die neue Platine (Bild 4). Auf dieser Platine kann man eine Menge an Bauteilen ausfindig machen, die aus der 1570, im Gegensatz zur 1541, wiederum ein komplett neues Gerät machen. Gleich geblieben sind die, schon in der 1541 enthaltenen Chips, Mikropro-

Befehl	Wirkung
NEW	Formatieren einer Diskette
INITIALIZE	Initialisieren einer Diskette
VALIDATE	Wiederherstellen der BAM
COPY	Kopieren von Files
RENAME	Umbenennen eines Files
SCRATCH	Löschen eines Files
POSITION	Setzen des Recordzeigers
B-R	Lesen eines Sektors
B-W	Schreiben eines Sektors
B-A	Belegen eines Sektors in der BAM
B-F	Freigeben eines Sektors in der BAM
B-E	Ausführen eines Programms in einem Sektor
B-P	Setzen des Pufferzeigers in einem Sektor
M-R	Lesen von Speicherinhalten des Laufwerks
M-W	Schreiben von Speicherinhalten
M-E	Starten eines Maschinenprogramms im Laufwerk
U1	Lesen eines Sektors
U2	Schreiben eines Sektors
U3 bis U8	Programmstarts im Benutzerpuffer
U:	Reset ausführen
&	Laden und Starten eines Maschinenprogramms
%R	Setzen der Leseversuche bei Fehlern
%S	Sektorabstand beim Schreiben setzen

Tabelle 4. Der Befehlssatz der 1551 gleicht stark dem der 1541

Speicherkapazität	174 848 (349 696) Byte total unter GCR 169 984 (339 968) Byte frei unter GCR 133 120 bis 204 800 Byte unter MFM (266 240 bis 409 600 bei 1571)
Directory-Einträge	144
Anzahl der Spuren	35 (70) unter GCR 40 (80) unter MFM
Anzahl der Sektoren	683 (1366) insgesamt unter GCR 664 (1328) frei unter GCR 17 bis 21 pro Spur unter GCR 200 bis 1040 (400 bis 2080) insg. unter MFM 5 bis 26 pro Spur unter MFM
Kapazität eines Sektors	256 Byte unter GCR (mit Linkpointern) 128, 256, 512, 1024 unter MFM
Aufzeichnungsformat	GCR und MFM
Aufzeichnungsrate	250 000 bit/s Abschnitt 1 unter GCR 266 664 bit/s Abschnitt 2 unter GCR 285 712 bit/s Abschnitt 3 unter GCR 307 688 bit/s Abschnitt 4 unter GCR
Bestückung	6502 Mikroprozessor VIA 65C22 (2) CIA 6526 2 KByte statisches RAM Logic Array als Diskcontroller für GCR WD 1770 Diskcontroller für MFM
Taktfrequenz	1 und 2 MHz
Übertragungsrate	1200 bit/s seriell (1541-Modus) bis 167 000 bit/s seriell (1570/71-Modus)

Tabelle 5. Die technischen Daten der 1570 (1571). In Klammern stehen jeweils die Werte für die 1571.

zessor 6502, zwei VIA 6522 und ein Logic-Array, das bei Commodore als Diskcontroller eingesetzt wird.

Auffällig sind die beiden VIA 6522. Hierbei handelt es sich nämlich um qualitativ sehr hochwertige 65C22 – also um die stromsparenden CMOS-Versionen. Der Prozessor ist in der Regel vom Typ 6502A, was ihn als 2-MHz-Version kennzeichnet.

Neu auf der Platine ist der schon erwähnte WD 1770 und ein CIA 6526. Der CIA findet auch im C 64 Verwendung, wobei CIA für »**C**omplex **I**nterface **A**dapter« steht.

Die 1570 enthält gegenüber der 1541 und auch der 1551 eine ganze Menge an zusätzlichen Neuheiten. Die Anlaufsteuerung beim Einlegen einer Diskette und eine Lichtschranke an der Spur-Null-Position sind dabei nur Nebensächlichkeiten.

Lesen von Fremdformaten

Der erste Unterschied wird deutlich, wenn man sich das ROM betrachtet, in dem das Betriebssystem der 1570 untergebracht ist. Im Gegensatz zur 1541 und zur 1551 handelt es sich hierbei um ein ROM mit 32 KByte Betriebssystem, also genau die doppelte Kapazität zur 1541.

In diesem ROM sind zum Beispiel die Routinen für den »normalen« Betrieb der 1570 vorhanden. Diese Routinen stellen im Prinzip das vollständige Betriebssystem der 1541 dar, weshalb die 1570 auch weit über 90 Prozent kompatibel zur 1541 ist.

Andererseits finden sich im ROM auch Teile zur Steuerung des Diskcontrollers WD 1770. Diese Erweiterung ist derart mächtig, daß die 1570 in die Lage versetzt wird, fast alle Disketten von bekannten Personal Computern zu lesen. Disketten von IBM, Kaypro, Osborne, Triumph Adler oder Compaq sind also kein Problem und können mit entsprechenden Programmen im Computer bequem verarbeitet werden.

Damit diese Diskettenformate überhaupt verwendet werden können, sind natürlich auch ein paar grundlegende Einrichtungen notwendig. Da ist zum einen die Möglichkeit zur Überwachung des Indexlochs einer Diskette. Hierfür hat die 1570 eine weitere Lichtschranke mit Infrarot-LED, die neben der Laufwerksachse angebracht ist.

Zur Verstärkung und Verarbeitung von Signalen dient in der 1570 ebenfalls die schon beschriebene Schaltung, die einen Hybridschaltkreis enthält.

Betrachtet man die 1570 als ganzes Gerät, so fällt auf, daß in diesem Diskettenlaufwerk eigentlich zwei verschiedene Geräte implementiert wurden.

Befehl	Wirkung
NEW	Formatieren einer Diskette
INITIALIZE	Initialisieren einer Diskette
VALIDATE	Wiederherstellen der BAM
COPY	Kopieren von Files
RENAME	Umbenennen eines Files
SCRATCH	Löschen eines Files
POSITION	Setzen des Recordzeigers
B-R	Lesen eines Sektors
B-W	Schreiben eines Sektors
B-A	Belegen eines Sektors in der BAM
B-F	Freigeben eines Sektors in der BAM
B-E	Ausführen eines Programms in einem Sektor
B-P	Setzen des Pufferzeigers in einem Sektor
M-R	Lesen von Speicherinhalten des Laufwerks
M-W	Schreiben von Speicherinhalten
M-E	Starten eines Maschinenprogramms im Laufwerk
User-Befehle:	
U1	Lesen eines Sektors
U2	Schreiben eines Sektors
U3 bis U8	Programmstarts im Benutzerpuffer
U:	Reset ausführen
&	Laden und Starten eines Maschinenprogramms
U0-Befehle:	
BURST READ	Lesen eines Sektors unter MFM oder GCR
BURST WRITE	Schreiben eines Sektors unter MFM oder GCR
INQUIRE DISK	Diskette initialisieren
FORMAT	Diskette beliebig formatieren
SECTOR INTERLEAVE	Sektorabstand setzen/holen
QUERY DISK	Diskettenformat identifizieren
INQUIRE STATUS	Burst-Status setzen/holen
FASTLOAD	Schnellader einschalten
U0>M0	1541-Modus einschalten
U0>M1	1570/71-Modus einschalten
U0>S	Sektorabstand beim Schreiben setzen
U0>R	Leseversuche bei Fehlern setzen
U0>T	Prüfsumme über ROM testen
U0>H	Diskettenseite setzen (nur 1571)
U0>	Gerätenummer des Laufwerks setzen

Tabelle 6. Die Befehlssätze der 1570 und 1571 weichen kaum voneinander ab. Lediglich die Befehle für zweiseitigen Diskettenbetrieb exisitieren nur auf der 1571.

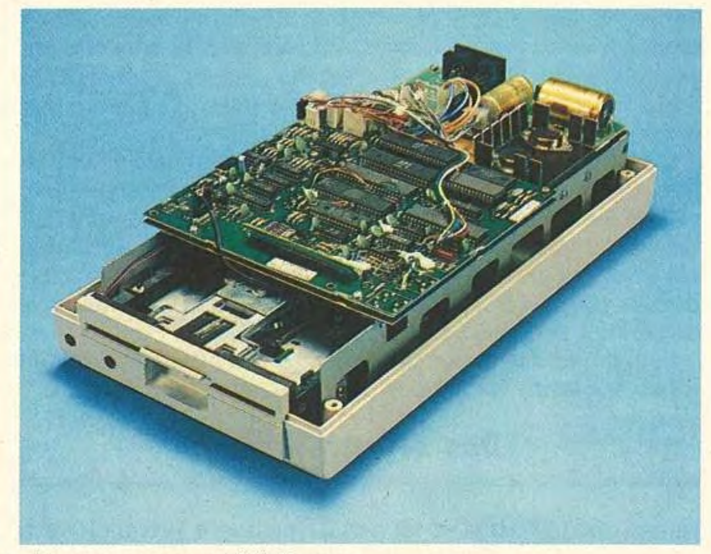


Bild 4. Die 1570 mit und ohne Gehäuse. Die Elektronik sieht ein wenig »zusammengewürfelt« aus.

Zum einen finden wir eine fast vollständige 1541 in der 1570 vor. Es wurden nur die Teile geändert, die für den erweiterten Betrieb der 1570 unbedingt notwendig sind. Wir sprechen in diesem Zusammenhang vom 1541-Modus, da diese Betriebsart über einen Befehl eingeschaltet werden kann.

Arbeitet man nicht im 1541-Modus, so befindet sich das Laufwerk immer im 1570-Modus. Hier steht ein sehr mächtiger und erweiterter Befehlssatz zur Verfügung (Tabelle 6), und das Laufwerk arbeitet nun mit einer Systemtaktfrequenz von 2 Megahertz. Jetzt wird auch deutlich, warum bei der Bestückung der Platine nur hochwertige Bauteile verwendet wurden. Die Bauteile müssen in der Lage sein, auch mit 2 MHz einwandfrei zu arbeiten.

Interessant an der 1570 ist aber sicherlich auch das neue Konzept des seriellen Bus. Arbeitet man im 1541-Modus, so arbeitet der serielle Bus wie von der 1541 her bekannt.

Im 1570-Modus schaltet sich jedoch der CIA 6526 in den Busbetrieb ein. Hier wird über das schnelle Schieberegister eine sehr viel höhere Übertragungsrate des seriellen Bus erreicht, die bis über 120000 bit/s erreichen kann.

Das erklärt auch die höhere Geschwindigkeit der 1570 gegenüber der 1541, die rein rechnerisch um den Faktor 100

höher ist. Diese Geschwindigkeitssteigerung kann jedoch normalerweise nur auf dem C 128 im C 128-Modus erreicht werden, da der C 64 den seriellen Bus anders beschaltet hat und somit nicht über ein angeschlossenes Schieberegister verfügt.

Sie können die 1570 dennoch am C 64 problemlos betreiben. Die meisten Programme laufen ohne Schwierigkeiten. Wollen Sie die hohe Geschwindigkeit der 1570 auch am C 64 oder am C 128 im C 64-Modus erreichen, so ist das Einlöten von zwei Kabeln und das Ändern des Betriebssystems erforderlich. Die 1570 wird dann zwischen sieben- und neunmal so schnell, wie im 1541-Modus.

Einen Beschleuniger dieser Art finden Sie übrigens im 64'er- Magazin, Ausgabe 9/86, als »Listing des Monats«.

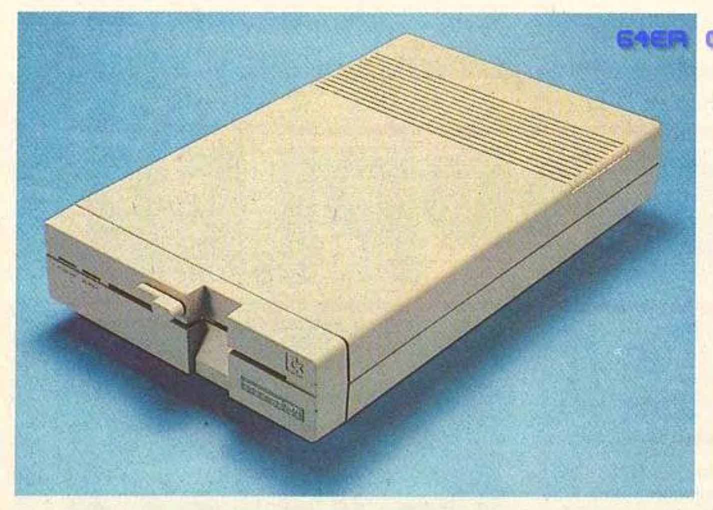
Das »Flaggschiff«

Neben der 1570 gibt es für den Commodore 128 auch noch ein anderes Diskettenlaufwerk, das die 1570 trotz deren Leistungsfähigkeit diesbezüglich noch übertrifft. Gemeint ist die 1571.

Dieses Laufwerk war ursprünglich schon vor der 1570 auf dem Markt. Commodore wollte jedoch noch eine »abgespeckte« Billigversion der 1571 produzieren, da die 1571 bei der Einführung nicht gerade preisgünstig zu haben war.

Prinzipiell gibt es nur zwei große Unterschiede zwischen beiden Diskettenlaufwerken, wobei der rein äußerliche Unterschied vernachlässigt werden soll.

Schraubt man die 1571 auf (Bild 5), so fällt sofort der komplett andere Aufbau im Inneren auf. Da ist zum einen das Laufwerk. Es handelt sich hierbei um ein komplett anderes Laufwerk als in der 1570. Aus dem großen Netztransformator, wie er in allen übrigen Laufwerken von Commodore zu finden ist, wurde ein kleines Netzteil eingebaut, das in einem Gitterkäfig untergebracht ist. Die Platine befindet sich unter dem Netzteil und ist somit vor zu großer Hitze geschützt.



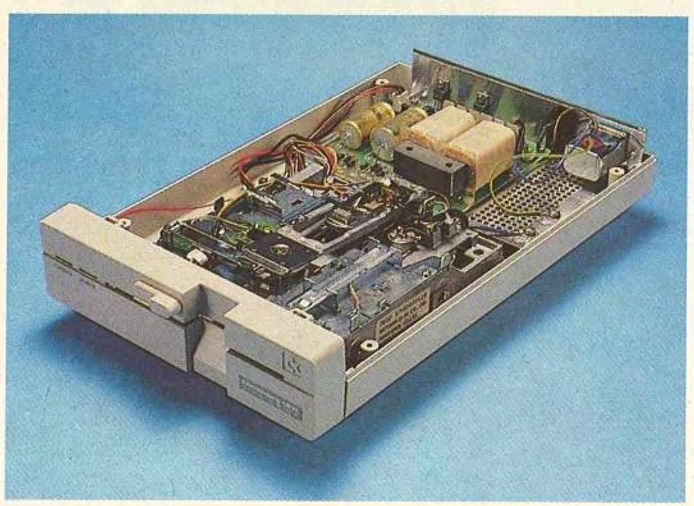


Bild 5. Die 1571 und deren Innenleben

Computer	Diskettenlaufwerk	Kompatibilität zu vorhandener Software
C 64	1541	100 Prozent
	1570	nahezu 98 Prozent
	1571	nahezu 95 Prozent
C 128	1541	ungeeignet für CP/M
	1570	nahezu 99 Prozent
	1571	100 Prozent
C 16, C 116, Plus/4	1551	100 Prozent (?)
	1541	100 Prozent
	1570	100 Prozent
	1571	nahezu 98 Prozent
VC 20	1541	100 Prozent
	1570	nahezu 99 Prozent
	1571	nahezu 98 Prozent

Tabelle 7. Anschlußmöglichkeiten der Commodore-Diskettenlaufwerke an die Commodore-Heimcomputer

Betrachtet man sich die Laufwerksmechanik genauer, so fällt sofort die Stärke der 1571 ins Auge. Es handelt sich hier um ein Laufwerk, das Disketten doppelseitig beschreiben kann, also zwei Schreib-/Leseköpfe besitzt. Die Folge dieser Einrichtung ist eine höhere Speicherkapazität (340 KByte) im Gegensatz zu allen anderen Laufwerken (170 KByte). Eine Diskette braucht also in Zukunft nicht mehr »gestanzt« und umgedreht zu werden. Sie wird einfach auf der 1571 formatiert und schon zeigt sich das Directory in ganz neuem Gewand: »1328 BLOCKS FREE«.

Wie die 1570, so besitzt auch die 1571 einen 1541-Modus. In diesem Modus arbeitet sie nach wie vor nur einseitig. Wir können also auch die 1571 problemlos an einem C 64 betreiben; allerdings mit dem Nachteil des Geschwindigkeitsverlustes.

Die technischen Daten zur 1571 finden Sie vergleichend zur 1570 in Tabelle 5. Der Befehlssatz ist in Tabelle 6 abgedruckt.

Welches Gerät an welchem Computer?

Zusammenfassend vielleicht noch einmal die Anschlußmöglichkeiten aller Diskettenlaufwerke von Commodore. In Tabelle 7 sehen Sie diese im Überblick.

Prinzipiell können Sie jedes Laufwerk (ausgenommen die 1551) an jeden Computer anschließen. Bei der 1541 am C 128 ergibt sich jedoch die Einschränkung, daß die Diskettenformate von zum Beispiel IBM oder Kaypro nicht gelesen werden können. Die 1551 kann nur am C 16, C 116 und Plus/4 angeschlossen werden. Die Kompatibilität ist bei den Laufwerken untereinander sehr hoch. Es gibt jedoch auch Ausnahmen. So befindet sich offensichtlich eine Produktreihe auf dem Markt, bei der die 1571 mit nur sehr wenigen 1541-Programmen läuft. Wir konnten noch nicht herausfinden, woran diese Inkompatibilität im einzelnen liegt. In der Regel laufen fast alle Programme der 1541 problemlos auch auf der 1570 und 1571. Auch die neue 1541 bildet hier keine Ausnahme.

(ks)

Wettlauf mit der Zeit

Mittlerweile gibt es derart viele Beschleunigungssysteme für die Diskettenstation 1541, daß der Käufer leicht den Überblick verliert. Der folgende Artikel hilft Ihnen bei der Kaufentscheidung weiter, indem er die aktuellsten Systeme einander gegenüberstellt.

Schnell sind sie alle, die Beschleuniger für die 1541. Die Palette reicht von einer Geschwindigkeitserhöhung um den Faktor 6 bis hin zum 200mal schnelleren »Turbolader«.

Der folgende Vergleichstest stellt einmal einige Systeme einander gegenüber. Es werden dabei sowohl die Stärken als auch die Schwächen der Speeder aufgedeckt.

Verlangt waren Beschleuniger, die Änderungen sowohl im Computer als auch im Laufwerk erfordern. Die Übertragung muß über ein zusätzliches Parallelkabel erfolgen, das Laufwerk und Computer miteinander verbindet.

Die Auswahlkriterien wurden aus folgenden Gründen getroffen:

1. Mit diesem Auswahlsystem erfassen wir garantiert die zur Zeit aktuellsten und verbreitetsten Systeme.
2. Die Beschleuniger, die die oben genannten Bedingungen erfüllen, sprechen alle ziemlich genau die gleiche Gruppe von Computeranwendern an und können aus diesem Grund leicht miteinander verglichen werden.
3. Wir haben es bei unseren Testkandidaten mit ziemlich aktuellen Produkten zu tun, die sich fast alle auf dem neuesten Stand ihrer Entwicklung befinden.
4. Der Aufwand beim Einbau der Systeme ist in etwa bei allen Speedern gleich, so daß sich Kaufentscheidungen nach den wichtigen Leistungsmerkmalen der Beschleunigungssysteme richten können.

Getestet wurden bei allen Systemen folgende Kriterien:

- Preis
- Lieferumfang
- Beschleunigung von Funktionen gegenüber einer Floppy 1541 in Originalform
- Verbesserung von Funktionen bezüglich des Originalgeräts
- Benutzerfreundlichkeit der einzelnen Systeme
- Kompatibilität der einzelnen Systeme zu Software, die sich augenblicklich auf dem Markt befindet
- Erweiterungen des Original-Betriebssystems des Computers und des Diskettenlaufwerks

Diese Testkriterien dürften sicherlich alle wichtigen Punkte einer Kaufentscheidung beinhalten.

Sie werden am Ende dieses Tests mit Bestimmtheit sagen können, welches System für Sie das richtige ist.

Nun aber zu unseren Kandidaten. Getestet wurden:

- Dolphin-Dos 2.0
- Professional DOS R4.0
- ProLogic-Dos
- SpeedDos Plus
- TurboAccess
- TurboTrans Plus 3.0

Die erste Teststufe beschäftigt sich mit der Geschwindigkeit des jeweiligen Systems, denn das ist schließlich der eigentliche Sinn dieser Erweiterungen.

Wir haben uns dazu spezielle Problemstellungen einfallen lassen, die insgesamt wohl die wichtigsten Aufgabengebiete eines Diskettenlaufwerks unter die Lupe nehmen.

Zu den Testaufgaben zählen natürlich die Messung der Lade- und Speichergeschwindigkeit. Zusätzlich wollen wir wissen, wie schnell eine Diskette mit VALIDATE »aufgeräumt« wird, in welcher Zeit der Speeder ein oder mehrere Programme von der Diskette löscht und wie hoch die Geschwindigkeit beim Umgang mit relativen Dateien ist. Natürlich spielt auch der Speicherplatz auf einer Diskette und die Dauer des Formatiervorgangs eine Rolle.

Neben diesen quantitativen Untersuchungen beschäftigen wir uns mit dem Aufbau eines Systems und seinem Befehlssatz in Floppystation und Computer. Wir untersuchen die Anwenderfreundlichkeit und die neuen Möglichkeiten eines Speeders. Es werden spezielle Eigenheiten sowohl positiver als auch negativer Art herausgestellt und der Gesamteindruck vom Lieferumfang bis hin zur Kompatibilität aufgezeigt.

Der »Urvater«

Zuerst soll uns einer der Urväter der Speeder mit Parallelkabel interessieren: SpeedDos Plus (Bild 1).

Wir haben SpeedDos Plus ganz bewußt in unsere Testreihe miteinbezogen, weil dieses Beschleunigungssystem infolge seiner frühen und jetzt auch noch andauernden Marktherrschaft Maßstäbe gesetzt hat. Viele Programme sind mittlerweile »SpeedDos-kompatibel«. Mehrere Speeder verwenden das »SpeedDos-Kabel« zur Datenübertragung, und oft hört man von Beschleunigern, die den »SpeedDos-Befehlssatz« und eine »SpeedDos-Tastenbelegung« besitzen.

Wir haben es also im wahrsten Sinne des Wortes mit einem »Urvater« der Parallelspeeder zu tun, der auch heute noch verkauft wird. Ein Vergleich mit den »Söhnen« dürfte aus diesem Grund sehr interessant sein, zumal SpeedDos Plus wegen seines Preis-/Leistungsverhältnisses nach wie vor sehr beliebt ist.

Der aktuelle Preis von SpeedDos Plus (die aktuelle und erweiterte Version des ursprünglichen SpeedDos) liegt bei 199,50 Mark. Dafür bekommen Sie eine kleine Platine, die im Diskettenlaufwerk anstelle des Original-DOS eingesteckt wird. Zusätzlich erhalten Sie ein Parallelkabel, das über einen Zwischensockel an die VIA 6522 angeschlossen wird. Im Computer wird ebenfalls eine kleine Platine untergebracht, die ein EPROM enthält. In dieses EPROM sind sowohl das

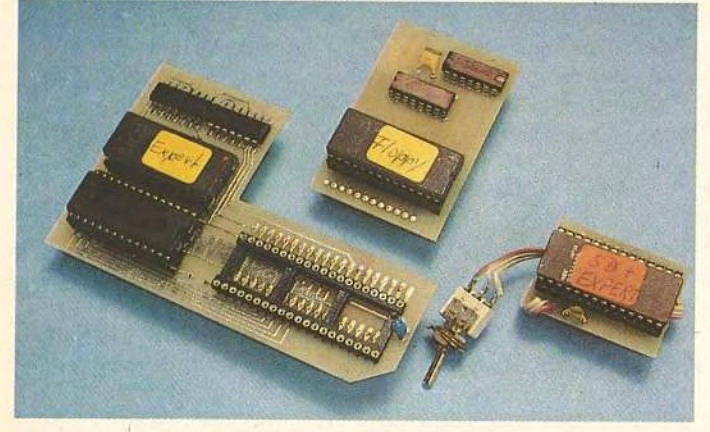


Bild 1. Die SpeedDos-Produktpalette

SpeedDos Plus als auch das Original-Commodore-Betriebssystem integriert. SpeedDos Plus kann dabei jederzeit über einen Schalter ausgeschaltet werden, und der Computer befindet sich im »Urzustand«.

Leider lag uns bei unserem Testmuster keine aktuelle Bedienungsanleitung zu SpeedDos Plus bei. Wir können über deren Qualität also keine gültige Aussage machen.

Am Testanfang stand der wohl am meisten verwendete Befehl LOAD. Wir haben dazu eine Diskette hergestellt, die 50 Files mit je einem Block Länge enthält. Das 51. File ist dann unser Testfile mit genau 202 Blöcken Länge.

Diese Diskette erlaubt uns einen objektiven Test der Ladezeit, die inklusive der Suche im Directory gemessen wird.

In der Werbung sieht man zuweilen, daß nur die »reine« Ladezeit ohne Suche im Directory angegeben wird. Diese Angabe ist, so meinen wir, irreführend, da gerade diese Directory-Suche eine ganze Menge Zeit in Anspruch nimmt. Wir zählen also das Suchen des Files im Directory zur Ladezeit dazu.

Zeiten mit Suche im Directory

Nun aber wieder zu SpeedDos Plus. Es erreichte bei unserer Testdiskette eine Ladezeit von 24 Sekunden für 202 Blöcke und Suche im Directory. Das entspricht im Gegensatz zum Original-Betriebssystem des C 64 und der Floppystation 1541 einer Verbesserung um den Faktor 5,4. Normalerweise benötigt der C 64 für die 202 Blöcke genau 130 Sekunden.

Bei diesem Geschwindigkeitsfaktor darf man nicht vergessen zu erwähnen, daß SpeedDos die Files normalerweise mit einem anderen Sektorabstand auf die Diskette schreibt. Wir verwendeten jedoch eine »Originaldiskette«. Wird ein File unter SpeedDos Plus neu auf die Diskette geschrieben, so kann ein Geschwindigkeitsfaktor von mehr als dem Zehnfachen der Originalgeschwindigkeit erreicht werden.

Natürlich darf auch der SAVE-Befehl nicht vergessen werden. SpeedDos Plus benötigte zum Speichern von 202 Blöcken auf die Diskette hinter 50 vorhandene Files genau 103 Sekunden. Das entspricht gegenüber der Geschwindigkeit des Original-DOS von 140 Sekunden einem Geschwindigkeitsfaktor von 1,4 beim Speichern.

Nach dem SAVEn wollten wir noch wissen, wie schnell ein File unter SpeedDos Plus mit SCRATCH gelöscht werden kann. Wir haben 27 Sekunden gemessen, wobei das Original-Betriebssystem mit 26 Sekunden etwa die gleiche Geschwindigkeit besitzt. Man kann also davon ausgehen, das die Löschroutine des DOS von SpeedDos Plus nicht geändert wurde. Das bestätigt uns auch unsere zweite Testmessung in Sachen SCRATCH. Hier werden 50 Files von einer Diskette gelöscht, die jeweils einen einzigen Block lang sind.

SpeedDos Plus benötigte für diese Aufgabe 72 Sekunden. Das Original-DOS schlägt mit 79 Sekunden zu Buche, wobei der Unterschied sicherlich auf die schnellere Bewegung des Schreib-/Lesekopfes unter SpeedDos zurückzuführen ist.

Um einen Wert für die Geschwindigkeit des Speeders bei der Behandlung von Dateien zu bekommen, haben wir das Erstellen einer relativen Datei gemessen. Es wurde dabei eine Datei eröffnet, die aus 200 Datensätzen besteht, wobei jeder Datensatz eine Länge von 140 Byte besitzt. Die Datei wurde bei unserem Test eröffnet und sämtliche 200 Datensätze angelegt. SpeedDos Plus benötigte für diese Aufgabe 59 Sekunden. Das Original-DOS der Floppy 1541 erreicht mit 60 Sekunden praktisch die gleiche Geschwindigkeit.

Auch im Löschen der relativen Datei waren sich SpeedDos Plus und das Original-Betriebssystem »einig«. Hier benötigten beide genau 16 Sekunden.

SpeedDos Plus

Testaufgaben:	gemessene Zeiten:
LOAD (202 Blöcke)	24 Sekunden
SAVE (202 Blöcke)	103 Sekunden
SCRATCH (202 Blöcke)	27 Sekunden
SCRATCH (REL-Datei)	16 Sekunden
SCRATCH (50 Files à 1 Block)	72 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)	59 Sekunden
Diskette validieren	168 Sekunden
Diskette formatieren (35 Spuren)	23 Sekunden
(40 Spuren)	–

Technische Daten des Beschleunigungssystems:	
Preis:	199,50 Mark
Computertyp:	C 64
Diskettenlaufwerk:	1541
Kapazität der Diskette (Spuren)	35 Spuren
(Sektoren insgesamt)	664 Sektoren
Speicher in der Floppy	2 KByte
Anschluß des Parallelkabels	User-Port
Taktfrequenz des Speeders	1 MHz
Speeder mit Schalter abschaltbar?	
Floppy	nein
Computer	ja

Wichtige zusätzliche Funktionen gegenüber dem Original-Betriebssystem:
- Centronics-kompatible Schnittstelle eingebaut (User-Port)
- Directoryanzeige ohne Programmverlust
- Funktionstasten belegt
- Hardcopy von Textbildschirm
- eingebauter Maschinensprache-Monitor
- OLD-Befehl zum Zurückholen eines gelöschten Basic-Programms

Tabelle 1. Testergebnisse und Daten von SpeedDos Plus

Die bisherigen Tests untersuchten die Floppystation jeweils unter durchschnittlichen Bedingungen, wie sie im Betrieb oft vorkommen. Wir wollten jedoch auch eine Extremsituation simulieren und so ließen wir alle Kandidaten eine vollständig gefüllte Diskette (»0 BLOCKS FREE« und 144 Einträge im Directory) mit dem Befehl VALIDATE »aufräumen«, wobei einzelne Files auf dieser Testdiskette vollkommen verstreut angeordnet waren.

Das Original-Betriebssystem der Floppy 1541 benötigte für diesen Durchlauf sage und schreibe 230 Sekunden (das sind fast vier Minuten).

Monitor eingebaut

SpeedDos Plus schlug sich hier wacker mit nur 168 Sekunden. Das sind gut zwei Drittel der Zeit des Original-DOS.

Natürlich durfte bei unseren Messungen auch das Formatieren einer Diskette nicht zu kurz kommen. Die Dauer dieses Vorgangs ist wohl das zweithäufigste Ärgernis, das einem Commodore-Anwender widerfährt.

SpeedDos Plus arbeitet auf der Diskette mit 35 Spuren und benötigt zum Formatieren 23 Sekunden. Zum Vergleich: Das Original-Commodore-DOS läßt sich für den gleichen Vorgang gut 85 Sekunden Zeit.

Von der Geschwindigkeit der 1541 einmal abgesehen, bietet SpeedDos Plus aber noch weitere Vorzüge. So beinhaltet das Betriebssystem zwar keine RS232-Routinen mehr, dafür existiert aber beispielsweise ein eingebauter kleiner Maschinensprachemonitor. Am User-Port wurde softwaremäßig

eine Centronics-kompatible Schnittstelle herausgeführt, an die jeder handelsübliche Drucker mit einer entsprechenden Schnittstelle angeschlossen werden kann (zum Beispiel Epson, Fujitsu, Star und andere). Viele auf dem Markt vorhandene Programme für Drucker unterstützen diese Schnittstelle am User-Port oder haben sogar eine eigene eingebaut, so daß ein Drucker-Interface für den seriellen Bus bei den meisten Anwendungen nicht mehr nötig ist. Lediglich wenn Sie Ausdrucke mit den Commodore-spezifischen Sonderzeichen anfertigen wollen, müssen Sie auf ein entsprechendes Interface zurückgreifen.

Kabelprobleme

Einen einzigen Nachteil hat die Centronics-kompatible Schnittstelle am User-Port aber dennoch: Wie schon oben erwähnt, wird am User-Port auch das Parallelkabel für das Diskettenlaufwerk angeschlossen. Sie haben also unter Umständen zwei Stecker für einen Anschluß. Um den Drucker und die Floppystation gleichzeitig betreiben zu können, ist ein Adapter notwendig, der am User-Port zwei Anschlüsse zur Verfügung stellt.

Zusätzlich zu den schon erwähnten Erweiterungen stellt SpeedDos Plus noch ein paar nützliche Tastenfunktionen zur Verfügung. So sind sämtliche Funktionstasten mit wichtigen Befehlen belegt, wobei auch das softwaremäßige Abschalten von SpeedDos Plus möglich ist. Die Tabelle 1 zeigt Ihnen noch einmal alle Testdaten und die wichtigsten Funktionen auf einen Blick.

TurboAccess

Testaufgaben:		gemessene Zeiten:
LOAD (202 Blöcke)		23 Sekunden
SAVE (202 Blöcke)		103 Sekunden
SCRATCH (202 Blöcke)		25 Sekunden
SCRATCH (REL-Datei)		14 Sekunden
SCRATCH (50 Files à 1 Block)		67 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)		57 Sekunden
Diskette validieren		155 Sekunden
Diskette formatieren	(35 Spuren)	18 Sekunden
	(40 Spuren)	20 Sekunden

Technische Daten des Beschleunigungssystems:		
Preis:		199 Mark
Computertyp:		C 64, C 128
Diskettenlaufwerk:		1541
Kapazität der Diskette	(Spuren)	35/40 Spuren
	(Sektoren insgesamt)	664/749 Sektoren
Speicher in der Floppy		2 KByte
Anschluß des Parallelkabels		Expansion-Port
Taktfrequenz des Speeders		1 MHz
Speeder mit Schalter abschaltbar?		
	Floppy	ja
	Computer	ja

Wichtige zusätzliche Funktionen gegenüber dem Original-Betriebssystem:
- Centronics-kompatible Schnittstelle eingebaut (User-Port)
- Directoryanzeige ohne Programmverlust
- Hardcopy von Textbildschirm
- OLD-Befehl zum Zurückholen eines gelöschten Basic-Programms
- verschiedene Reset-Routinen zum Überspringen der Modul-Kennung und zum Erhalten der Variablen und der Basic-Programme

Tabelle 2. Das leistet TurboAccess

Etwa zur gleichen Zeit wie SpeedDos Plus kam auch ein anderes Beschleunigungssystem auf den Markt: TurboAccess.

TurboAccess weist von der Konzeption her ein paar größere Unterschiede zu SpeedDos Plus auf. Das wird schon am Lieferumfang deutlich. Für den Preis von 199 Mark bekommt der Anwender eine Platine für das Diskettenlaufwerk, die durch ein Kabel mit einer weiteren Platine für den Expansion-Port des Computers verbunden ist. Der User-Port bleibt bei TurboAccess also frei. Im Computer muß unter das Betriebssystem ein kleiner Adaptersockel gesetzt werden, der über ein Kabel mit der Platine im Expansion-Port verbunden ist.

Auf der Erweiterungsplatine am Computer befindet sich unter anderem ein Schalter. Mit diesem kann TurboAccess hardwaremäßig ausgeschaltet und das Original-Betriebssystem sowohl im Diskettenlaufwerk als auch im Computer wieder eingeschaltet werden.

Der Konkurrent

TurboAccess wurde natürlich unter den gleichen Testbedingungen untersucht wie SpeedDos Plus. Dabei zeigten sich leichte Zeitvorteile für TurboAccess. Das Laden eines 202-Block-Files mit 50 Programmen, die im Directory davor stehen, dauert 23 Sekunden. Gespeichert werden die 202 Blöcke in 103 Sekunden. Nach dem Speichern kam das Löschen. Hier benötigte TurboAccess 27 Sekunden. Das Erstellen unserer relativen Testdatei dauerte alles in allem 57 Sekunden. Gelöscht wurde die Datei in nur 14 Sekunden. Der Löschtest mit 50 1-Block-Files wurde von TurboAccess in nur 67 Sekunden bewältigt, und sogar für das Validieren der Testdiskette benötigte TurboAccess weniger Zeit als SpeedDos Plus: nämlich ganze 155 Sekunden. Das Formatieren geht bei TurboAccess trotz eines eingebauten Verify sehr flott. Es dauerte in unserem Test nicht länger als 18 Sekunden und dürfte damit allen Ansprüchen genügen. So schnell sind normalerweise noch nicht einmal Personal Computer. Das Laden von unserer Testdiskette bedeutet übrigens, wie schon bei SpeedDos Plus, nicht, daß es nicht noch schneller geht. Auch bei TurboAccess wird nämlich beim Speichern der Sektorabstand auf der Diskette verändert, so daß Files, die unter TurboAccess gespeichert wurden, noch sehr viel schneller wieder geladen werden können. Der Geschwindigkeitsfaktor beträgt auch hier etwa 10.

Das Betriebssystem von TurboAccess beinhaltet zwar ähnliche Funktionen wie SpeedDos Plus. Es wurde jedoch völlig anders konzipiert. So verzichteten die Entwickler von TurboAccess bewußt auf die Belegung der Funktionstasten, da gerade diese praktische Belegung mit manchen Programmen Probleme bekommt. Bei TurboAccess werden die Zusatzfunktionen allesamt über die Tastenkombination <CTRL+Taste> aufgerufen.

Wie schon SpeedDos Plus, so enthält auch TurboAccess eine Centronics-kompatible Schnittstelle am User-Port. Der Anschluß eines Druckers macht hier jedoch keine Probleme, da die Floppystation, wie schon erwähnt, am Expansion-Port des Computers angeschlossen wird. Auch Befehlserweiterungen, wie Laden aus dem Directory oder der OLD-Befehl, der ein gelöschtes Basic-Programm zurückholt, sind in TurboAccess eingebaut. Der Mini-Maschinensprachemonitor von SpeedDos Plus fehlt in der Grundversion von TurboAccess. Das ist Absicht, da die Entwickler von TurboAccess auf vorhandene RS232-Routinen im Betriebssystem Wert legen und diese sonst hätten entfernen müssen.

Die Hardware von TurboAccess ist ungleich aufwendiger als die von SpeedDos Plus und macht auch qualitätsmäßig einen besseren Eindruck. Die Lösung mit dem Übertragungskabel am Expansion-Port ist technisch auf jeden Fall besser.

Sie hat sich jedoch leider auf dem Markt nicht so durchsetzen können wie die SpeedDos-Variante.

Technisch gesehen hat TurboAccess also einen Vorteil gegenüber SpeedDos Plus. Das betrifft sowohl die höhere Qualität der Hardware als auch die Möglichkeit, ein zweites Laufwerk im Parallelbetrieb an das vorhandene System anzuschließen. Die Bedienerfreundlichkeit des Systems ist jedoch bei SpeedDos besser. Hier macht sich die Belegung der Funktionstasten bemerkbar, an die man sich sehr schnell gewöhnt und die man auf anderen Computersystemen leicht vermißt. Die Tabelle 2 zeigt Ihnen alle wichtigen Testdaten und Funktionen von TurboAccess.

Das ausgereifte System

Unser nächster Kandidat ist Dolphindos 2.0 (Bild 2). Hierbei handelt es sich, wie auch bei den folgenden Beschleunigern, um ein System der zweiten Generation. Gemeint ist damit der technische Aufwand, der mit den Speedern getrieben wird.

Während SpeedDos Plus und TurboAccess lediglich ein neues Betriebssystem und ein Zusatzkabel in der Floppystation unterbringen, arbeiten die neuen Speeder mit zusätzlichem Speicher in der Floppystation und äußerst ausgefeilten Tricks zur Beschleunigung sämtlicher Funktionen des Laufwerks.

Dolphindos 2.0 ist in dieser Hinsicht eine kleine Revolution auf dem Markt der Floppy-Speeder. Es kostet als Komplettsystem nur 198 Mark. Dieses »nur« in der Preisangabe wird sich bei den anschließenden Besprechungen klären, es ist jedoch in jedem Fall angebracht.

Als Komplettsystem bekommen Sie bei Dolphindos 2.0 eine Platine, die in die Floppystation eingesteckt wird. Auf dieser Platine befinden sich unter anderem zwei Steckanschlüsse für Parallelkabel. Dolphindos 2.0 erlaubt es also dem Anwender, zwei Laufwerke im Parallelbetrieb zu verwenden, wobei das eine mit der Gerätenummer 8 und das andere mit der 9 angesprochen wird.

Für den Computer erhält der Anwender ein EPROM auf einer Adapterplatine, das wahlweise ein Umschalten zwischen dem Original-Betriebssystem und dem Dolphindos 2.0 erlaubt. Es ist auf diese Art und Weise möglich, den Originalzustand des Computers wiederherzustellen, wenn das nötig werden sollte. In der Floppystation existiert ein solcher Umschalter ebenfalls, so daß auch das Diskettenlaufwerk auf Originalbetrieb zurückgeschaltet werden kann.

Zusätzlich zur Hardware, die außerdem noch aus einem Verbindungskabel besteht, das am User-Port eingesteckt wird, bekommt der Käufer noch ein kleines Heftchen, das auf 20 Seiten ausführlich mit der Bedienung von Dolphindos 2.0 vertraut macht. Auch der Einbau des Systems ist sorgfältig beschrieben. An der Anleitung gibt es vom Inhalt her also nichts Wesentliches zu bemängeln. Lediglich zusammengeheftet könnte es noch sein, da es einem sonst passieren kann, daß sich alle Teile der Anleitung in unterschiedlicher Himmelsrichtung voneinander »entfernen«.

Auf einer beigefügten Diskette bekommt der Anwender zusätzlich noch zwei Kopierprogramme, die die Geschwindigkeit von Dolphindos 2.0 ausnutzen und vor allem einzelne Files blitzschnell kopieren.

Nun aber zu den Meßergebnissen mit unseren Testdisketten. Es zeigt sich hier natürlich sofort, mit was für einem »Kaliber« wir es bei den Speedern der neueren Generation zu tun haben. Da diese Beschleunigungssysteme den Speicher der Floppystation erweitern, sind sie in der Lage, jeweils eine gesamte Spur einer Diskette auf einmal einzulesen. Daß diese Maßnahme stark zur Beschleunigung beiträgt, vor allem wenn man weiß, daß zusätzlich auch das ROM um acht KByte Speicher erweitert wurde, ist unschwer einzusehen.

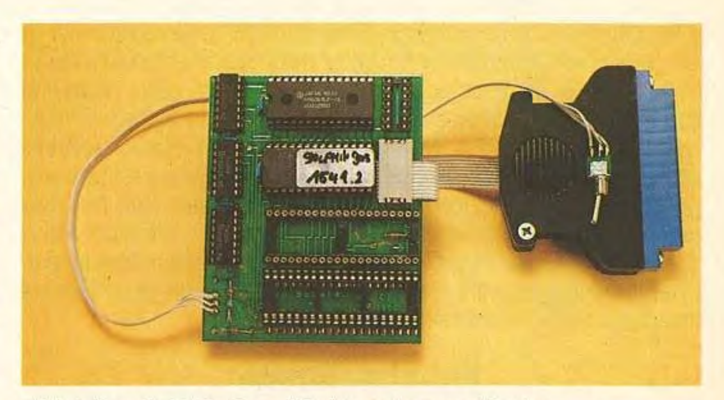


Bild 2. Das Dolphindos – künftig mit neuer Platine

So dauert das Laden von 202 Blöcken inklusive Suche im Directory auch nur mehr 5,5 Sekunden. Es macht einem direkt Spaß, der Diskettenstation bei der Arbeit zuzusehen. Wehmütig wird man erst, wenn ein »gepacktes« Programm in den Speicher des Computers geladen wird. Bei diesen »zusammengestauchten« Programmen dauert das »Entpacken« in der Regel länger als das Laden von der Diskette.

Auch beim Speichern konnte Dolphindos 2.0 voll und ganz überzeugen. In nur 10 Sekunden werden 202 Blöcke auf die Diskette gespeichert. Wir erinnern uns: Das Original-Betriebssystem des C64 benötigt ganze 140 Sekunden. Dolphindos 2.0 ist also 14mal schneller.

Das Löschen der 202 Blöcke von der Diskette ist ebenfalls kein Problem. In nur 4,5 Sekunden ist von den 202 Blöcken nichts mehr übrig und die Diskette strahlt uns wieder ihr »614 BLOCKS FREE« entgegen. (Es befinden sich noch 50 Files mit je einem Block Länge auf der Diskette, die wir vor den 202 Blöcken gespeichert hatten, um die Suche im Directory als Zeitfaktor miteinbeziehen zu können.)

Auch das Erstellen einer relativen Datei mit 200 Datensätzen à 140 Zeichen geht unter Dolphindos 2.0 erstaunlich schnell. Nach 20 Sekunden war die ganze Arbeit getan, und unsere relative Datei war angelegt.

Zum Löschen der relativen Datei waren ganze 3,5 Sekunden notwendig. Direkt im Anschluß daran haben wir Dolphindos 2.0 auch 50 Files mit je einem Block Länge löschen lassen. Das Ergebnis ist zwar im Vergleich nicht so spektakulär, es kann sich jedoch sehen lassen: 51 Sekunden für den gesamten Löschvorgang.

Den Vogel abgeschossen hat Dolphindos 2.0 im Test der Validate-Geschwindigkeit. Wir hatten dafür eine Diskette vorbereitet, die bis auf das letzte Byte gefüllt war. Das heißt: 144 Einträge im Directory und »0 BLOCKS FREE«, wobei viele Files kreuz und quer auf der Diskette standen, um viele Bewegungen des Schreib-/Lesekopfes zu provozieren.

Dolphindos 2.0 hat diese Diskette in sage und schreibe 20 Sekunden »abgegrast«. Das ist mit Abstand die schnellste Zeit, die ein Speeder auf der Diskette bisher erreicht hat. Zum Vergleich: Das Original-Betriebssystem der Floppy 1541 war fast vier (4!) Minuten am Arbeiten.

35 und 40 Spuren

Eine weitere Besonderheit von Dolphindos 2.0 ist die Tatsache, daß der Anwender zwischen dem Diskettenformat mit 35 oder dem Format mit 40 Spuren wählen kann. Läßt man eine Diskette mit 35 Spuren formatieren, so ist das nach 19 Sekunden geschehen, und die üblichen 166 KByte Speicherplatz stehen dem Anwender zur Verfügung.

Bei einer Formatierung auf 40 Spuren hingegen ist Dolphindos 2.0 21 Sekunden beschäftigt. Danach »lächeln« einen jedoch »749 BLOCKS FREE« an. Das entspricht einer

Grund genug fürs neue 64'er:

Text-verarbeitungs-system und Daten-fernübertragung:

Was Sie können und leisten müssen.

In der Oktober-Ausgabe stellen wir vor,

1. Die neue Rubrik Lexikon »Der Fachbegriffe und Literaturhinweise«

- Fachbegriffe und Computersprache werden ausführlich erklärt.
- Computer-Neulinge erfahren alles, was sie zum Einstieg und für die Anwendung wissen müssen.

2. einen interessanten Basic-Kurs

... außerdem lesen Sie:

■ Den großen Mailbox-Vergleichstest der Mailboxen in Deutschland. ■ Wir stellen die interessantesten Mailboxen Amerikas vor. ■ Sie erhalten eine ausführliche Marktübersicht von Akustikkopplern und Moden. ■ Akustikkoppler im Test: Was Sie leisten. ■ Datenfernübertragung: aber wie? ■ Das Listing des Monats: Der Soundmonitor. ■ Anwendung des Monats: Ein Sprachdigitizer zum Nachbauen. ■ Tips und Tricks zum Mastertext »Vizawrite« und »Glos« ■ Der große Grafikprogramm-Test.

Falls Sie »64'er« noch nicht regelmäßig beziehen, sichern Sie sich jetzt Ihr persönliches Abonnement und nutzen die damit verbundenen Vorteile: ■ Sie beziehen »64'er« ohne Mehrkosten bequem per Post frei Haus ■ Sie haben Ihr »64'er« bereits bei sich zu Hause — noch bevor Sie es bei Ihrem Zeitschriftenhändler kaufen können. ■ Sie sind sicher, keine Ausgabe zu versäumen.

Sie erhalten — wenn Sie zur Anforderung den nebenstehenden Gutschein verwenden — auf alle Fälle die neueste Ausgabe als Probeheft unverbindlich und kostenlos.

Gutschein

FÜR EIN KOSTENLOSES PROBEEXEMPLAR DES 64'er-MAGAZINS

JA, ich möchte »64'er«, das Magazin für Computerfans, kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »64'er« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte »64'er« dann regelmäßig frei Haus per Post und bezahle pro Jahr nur DM 78,— (Ausland auf Anfrage).

Vorname, Name

Straße — PLZ, Ort

Datum — 1. Unterschrift

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

Datum — 2. Unterschrift

Gutschein ausfüllen, ausschneiden, in ein Kuvert stecken und absenden an:
Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar

64S986

Speicherkapazität von 187,25 KByte bei voller Datensicherheit. Schließlich gibt es offiziell keine 35-Spur-Laufwerke, sondern lediglich 40-Spur-Laufwerke, und dazu zählt auch die 1541-Floppystation. Der Anwender braucht also nicht zu befürchten, daß sich die Floppy 1541 mit dieser Anzahl an Spuren »übernimmt«.

Super-Betriebssystem

Wir haben von Dolphindos 2.0 zwar jetzt allerlei Sensationszeiten gehört; das Beste an diesem Betriebssystem ist aber sicherlich das Kernel im Computer. Hier wurde lange entwickelt und herumgefeilt, mit dem Ergebnis, daß wir in diesem Testbericht bei Dolphindos 2.0 sicherlich von einem ausgereiften System sprechen können.

Das Wort ausgereift steht dabei jedoch nicht nur für die Fehlerfreiheit, es soll vielmehr die sehr durchdachte Bedienungsfreundlichkeit zum Ausdruck bringen. Kein anderes Betriebssystem konnte uns in dieser Hinsicht bisher so begeistern.

Das fängt schon damit an, daß der Käufer bei Dolphindos

Dolphindos 2.0

Testaufgaben:		gemessene Zeiten:
LOAD (202 Blöcke)		5.5 Sekunden
SAVE (202 Blöcke)		10 Sekunden
SCRATCH (202 Blöcke)		4.5 Sekunden
SCRATCH (REL-Datei)		3.5 Sekunden
SCRATCH (50 Files à 1 Block)		3.5 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)		20 Sekunden
Diskette validieren		20 Sekunden
Diskette formatieren	(35 Spuren)	19 Sekunden
	(40 Spuren)	21 Sekunden

Technische Daten des Beschleunigungssystems:

Preis:		198 Mark
Computertyp:		C64, C128
Diskettenlaufwerk:		1541
Kapazität der Diskette (Spuren)		35/40 Spuren
	(Sektoren insgesamt)	664/749 Sektoren
Speicher in der Floppy		10 KByte
Anschluß des Parallelkabels		User-Port
Taktfrequenz des Speeders		1 MHz
Speeder mit Schalter abschaltbar?		
	Floppy	ja
	Computer	ja

Wichtige zusätzliche Funktionen gegenüber dem Original-Betriebssystem:
- Centronics-kompatible Schnittstelle eingebaut (User-Port)
- Directoryanzeige ohne Programmverlust
- Hardcopy vom Textbildschirm
- OLD-Befehl zum Zurückholen eines gelöschten Basic-Programms
- verschiedene Reset-Routinen zum Überspringen der Modul-Kennung und zum Erhalten der Variablen und der Basic-Programme
- Maschinensprache-Monitor eingebaut
- Funktionstasten belegt (12 Funktionstasten)
- Bildschirm-Reset auf Tastendruck
- Maschinenprogramme mit <SHIFT+RUN/STOP> laden und starten
- Modifizierter SYS-Befehl mit möglichen Hex-Eingaben
- Umrechnung zwischen verschiedenen Zahlensystemen
- (Abschaltbare) Auto-Repeatfunktion aller Tasten
- Speeder in Stufen zurückschaltbar (für höhere Kompatibilität)
- Freie Belegung der Funktionstasten durch Anwender möglich

Tabelle 3. Das Preis-/Leistungswunder Dolphindos 2.0

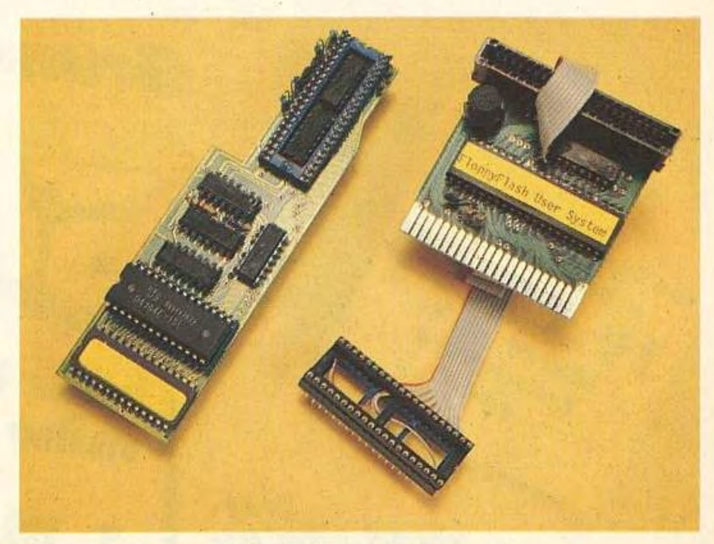


Bild 3. Professional Dos R4.0, das Aufbausystem. Hier bekommen alte Speeder neuen Glanz.

2.0 nicht befürchten muß, heute ein System zu kaufen, das morgen schon veraltet ist. Während andere Hersteller ihre Systeme noch weiterentwickeln, wobei sich der Käufer unter Umständen öfters um Updates bemühen muß, erhält er mit Dolphindos 2.0 ein ausgereiftes System: Dolphindos 2.0 ist laut Firmenaussage fertig entwickelt.

Arbeitet man mit Dolphindos 2.0, so fällt sofort die Ähnlichkeit mit SpeedDos Plus ins Auge. In der Tat haben sich die Entwickler in den Grundzügen an SpeedDos Plus orientiert, da SpeedDos Plus sehr verbreitet ist und einige Programme sogar nur unter diesem System lauffähig sind. Ein Beispiel ist »FCOPY III«. Dieses schnelle Backup-Programm verwendet spezielle Einsprungadressen von SpeedDos Plus, die absichtlich unter Dolphindos 2.0 nachempfunden wurden.

Dolphindos 2.0 bietet aber noch einiges mehr. Die Centronics-kompatible Schnittstelle am User-Port ist dabei fast nicht mehr der Erwähnung wert; sie gehört bei Speedern mittlerweile zur Standardausrüstung.

Vielfalt mit Funktionstasten

Es existiert ein eingebauter Maschinensprache-Monitor zum Ansehen, Laden und Speichern von Speicherbereichen und eine Funktionstastenbelegung. Neben der Belegung von 12 (!) Funktionstasten (die Funktionstasten 9 bis 12 werden zusammen mit der Commodore-Taste (CBM) aufgerufen) sind auch noch mehrere Tasten zusammen mit der <CTRL>-Taste belegt. Hier existieren Funktionen wie Bildschirm-Reset, Repeatfunktion der Tasten an/aus, Bildschirmzeile vor/hinter dem Cursor löschen und so weiter. Nebenbei wurden auch noch zusätzliche Befehle im Betriebssystem integriert. Dabei sind zum Beispiel die Umrechnung von Zahlensystemen oder der OLD-Befehl, der das Zurückholen eines Basic-Programms nach dem Löschen erlaubt.

Da der Kassettenpuffer nicht mehr benötigt wird, legte man unter Dolphindos 2.0 die Belegung der Funktionstasten in diesen Bereich. Es wird dem Anwender so ermöglicht, die Tasten ganz bequem umzudefinieren. Hat man es mit Programmen zu tun, die den Kassettenpuffer in gelöschtem Zustand benötigen, so genügt die Tastenkombination <CTRL+X>, und die Funktionstasten werden abgeschaltet und der Kassettenpuffer mit Nullen gefüllt. Danach sind die betreffenden Programme garantiert lauffähig. Tabelle 3 zeigt noch einmal die Ergebnisse unseres Tests.

Sie sehen also, daß die Entwickler an Dolphindos 2.0 intensiv gearbeitet haben, oder kennen Sie einen anderen

Floppy-Speeder, bei dem Sie Maschinenprogramme mit <SHIFT+RUN/STOP> laden und starten (!) können?

Neben Speedern der ersten Generation und Speedern der zweiten Generation haben wir Ihnen auch noch etwas Besonderes in unserem Vergleichstest anzubieten. Es handelt sich um einen Speeder der zweiten Generation für Speeder der ersten Generation.

Der Speeder für den Speeder

Das Professional Dos R4.0 (Bild 3) nutzt die Tatsache für sich aus, daß die neuen Beschleuniger für die Floppystation in der Regel um einiges schneller und komfortabler sind als die älteren Systeme, wie zum Beispiel SpeedDos Plus oder Turbo-Access.

Es wird dem Anwender daher angeboten, sein altes System aufzurüsten oder gleich ein neues Komplettsystem zu kaufen. Dabei sind im Augenblick ein Aufrüstsatz für SpeedDos und Floppy-Flash erhältlich. Ein Aufrüstsatz für TurboAccess ist nach Auskunft von Mikrotronic ebenfalls zu haben.

Es stand uns für unseren Test das SpeedDos Plus-System zur Verfügung. Die Aufrüstsätze kosten für SpeedDos Plus und Floppy-Flash jeweils 169 Mark für die 35-Spur-Version und 189 Mark für die 35/40-Spur-Version. Die Erweiterung für TurboAccess ist jeweils um 10 Mark teurer.

Bei dem System erhalten Sie eine Platine für die Floppystation und ein neues Kernel für den Computer, in dem sowohl das Original- als auch das neue Betriebssystem enthalten sind. Zum Lieferumfang bei der Floppy-Flash-Version gehört außerdem eine Diskette, die ein schnelles File-Kopierprogramm und zwei EPROM-Generatoren enthält. Die EPROM-Generatoren erlauben es, entweder ein Betriebssystem mit ein- oder eines mit ausgeschaltetem Bildschirm beim Laden in ein EPROM zu »brennen«. Das Betriebssystem mit ausgeschaltetem Bildschirm ist dabei in den Diskettenfunktionen noch geringfügig schneller. Getestet haben wir Professional Dos R4.0 mit eingeschaltetem Bildschirm, da sämtliche anderen Testkandidaten ebenfalls damit arbeiten.

Neben dem erwähnten Lieferumfang erhielten wir noch ein paar Seiten Anleitung, die leider nicht in Heftform vorhanden waren. Es zeigt sich hier, wie leider sehr häufig, daß bei einem der wichtigsten Punkte von Herstellern immer wieder an falscher Stelle gespart wird. In der Anleitung, die zumindest den Einbau des Systems gut beschreibt, finden Produktinformationen und Bezugsquellen fast mehr Platz als die eigentliche Beschreibung des Systems.

Nun aber zum Test von Professional Dos R4.0. Beim Laden von 202 Blöcken mit der Suche im Directory wurden 5,5 Sekunden gemessen. Das ist eine sehr gute Zeit, die deutlich den Unterschied älterer und neuerer Beschleunigungssysteme aufzeigt. Das Speichern des 202-Block-Testfiles dauerte mit 10 Sekunden genauso lang wie bei anderen neuen Systemen. Beim Löschen des Files ergab sich eine Meßzeit von 7,4 Sekunden.

Das Einrichten unserer relativen Testdatei dauerte unter Professional Dos R4.0 nur mehr 12 Sekunden. Dolphindos 2.0 wird hier also um mehr als 7 Sekunden unterboten. Zum Löschen der betreffenden Datei benötigte Professional Dos R4.0 5 Sekunden. Das Löschen der 50 Testdateien dauerte 59 Sekunden, womit sich Professional Dos R4.0 im Mittelfeld bewegte.

Unsere Spezialdiskette für den VALIDATE-Befehl ergab eine Zeit von 118 Sekunden für das »Aufräumen«. Fairerweise muß an dieser Stelle vielleicht gesagt werden, daß das Professional Dos R4.0 in der Lage ist, die Systemtaktfrequenz der Floppy 1541 auf 2 MHz zu erhöhen. Das passiert softwaremäßig immer dann, wenn zeitkritische Abläufe begonnen werden. Dolphindos 2.0 arbeitet grundsätzlich mit einem Megahertz und ist deshalb bei gewissen Funktionen zwangsläufig im Nachteil, wie auch alle anderen Speeder, die die Taktfrequenz der Diskettenstation nicht erhöhen.

Auch Professional Dos R4.0 ist in der Lage, entweder mit 35 oder mit 40 Spuren auf der Diskette zu arbeiten. Das Formatieren von 35 Spuren dauert dabei 21 Sekunden. Für 40 Spuren ist der Speeder 24 Sekunden lang beschäftigt. Auch hier stehen dann, wie bei Dolphindos 2.0 auch, 664 beziehungsweise 749 Blöcke zur Verfügung.

Die Funktionen im Betriebssystem des Computers sind bei Professional Dos R4.0 auf das Wesentliche beschränkt. Es existiert eine Centronics-kompatible Schnittstelle am User-Port und eine OLD-Funktion zum Zurückholen von gelöschten Basic-Programmen, eine Tastenkombination zum Starten einer Hardcopy des Textbildschirms und eine Belegung der Funktionstasten. Zusätzlich sind mehrere Reset-Routinen implementiert, die das Umgehen von Autostart-Programmen erlauben.

Das Professional DOS R4.0 besitzt zwar keinen eingebauten Mini-Maschinensprache-Monitor. Dafür sind die RS232-Routinen des Betriebssystems noch vorhanden. Lediglich die Routinen für den Betrieb einer Datasette wurden entfernt. Das dürfte vor allem für alle jene Anwender von Interesse sein, die sich mit DFÜ beschäftigen. Tabelle 4 zeigt noch einmal die Testergebnisse und eingebauten Funktionen auf einen Blick.

Professional Dos R4.0

Testaufgaben:	gemessene Zeiten:
LOAD (202 Blöcke)	5.5 Sekunden
SAVE (202 Blöcke)	10 Sekunden
SCRATCH (202 Blöcke)	7.4 Sekunden
SCRATCH (REL-Datei)	5 Sekunden
SCRATCH (50 Files à 1 Block)	59 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)	12 Sekunden
Diskette validieren	118 Sekunden
Diskette formatieren (35 Spuren)	21 Sekunden
(40 Spuren)	24 Sekunden

Technische Daten des Beschleunigungssystems:

Preis:	189 Mark
Computertyp:	C64, C128
Diskettenlaufwerk:	1541
Kapazität der Diskette (Spuren)	35/40 Spuren
(Sektoren insgesamt)	664/749 Sektoren
Speicher in der Floppy	10 KByte
Anschluß des Parallelkabels	User-Port
Taktfrequenz des Speeders	1/2 MHz
Speeder mit Schalter abschaltbar?	
Floppy	nein
Computer	ja

Wichtige zusätzliche Funktionen gegenüber dem Original-Betriebssystem:

- Centronics-kompatible Schnittstelle eingebaut (User-Port)
- Directoryanzeige ohne Programmverlust
- Maschinensprache-Monitor eingebaut
- Funktionstasten belegt
- Modifizierter SYS-Befehl mit möglichen Hex-Eingaben
- Umrechnung zwischen verschiedenen Zahlensystemen
- Speeder in Stufen zurückschaltbar (für höhere Kompatibilität)
- Umschaltung der Taktfrequenz in der Floppystation

Tabelle 4. Die Daten des Professional Dos R4.0

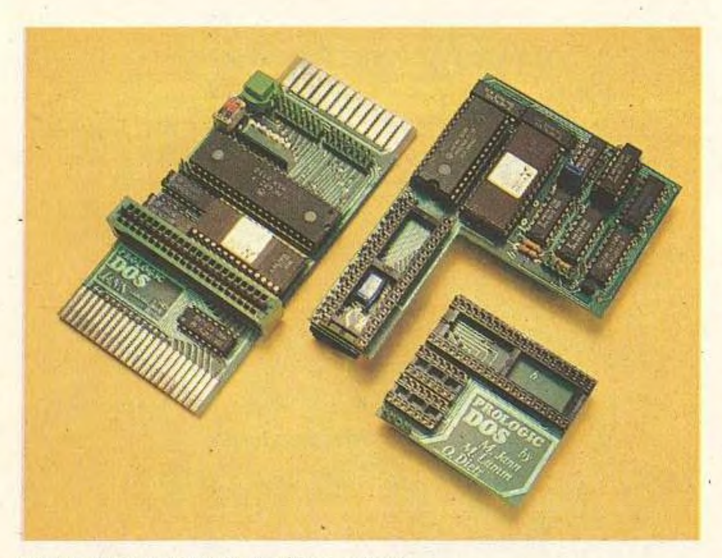


Bild 4. ProLogic-Dos, das Rennpferd

Der »Aufwendige«

Der nächste Kandidat unserer Testreihe ist ProLogic-Dos (Bild 4). Für 298 Mark bekommen Sie zwei Platinen, die in der Floppystation 1541 untergebracht werden. Dazu gibt es eine Platine, die in den Expansion-Port des Computers eingesteckt wird. Neben diesen drei Platinen erhält man ein ausführliches 30seitiges Handbuch, das von den bisher getesteten Produkten den besten Eindruck macht. Es erläutert ausführlich den Einbau und die Bedienung des Systems, wobei festzustellen ist, daß ProLogic-Dos infolge seiner höheren Komplexität (größerer Hardware-Aufwand) von allen Speedern am kompliziertesten einzubauen ist.

ProLogic-Dos arbeitet in der Floppy 1541 teilweise mit 2 MHz Taktfrequenz und ist deshalb natürlich auch technisch allen anderen bisher getesteten Systemen, mit Ausnahme des Professional Dos R4.0, überlegen. Man darf also auf die Testzeiten gespannt sein.

Das Laden eines Programms mit 202 Blöcken Länge dauert wie schon bei Dolphindos 2.0 und Professional Dos R4.0, 5,5 Sekunden. Zum Speichern des gleichen Programms benötigte ProLogic-Dos, wie schon Dolphindos 2.0, 10 Sekunden. Erstaunlicherweise ist ProLogic-Dos beim Löschen von 202 Blöcken langsamer als Dolphindos 2.0. Hier wurden ganze 7 Sekunden benötigt.

Das Bearbeiten unserer relativen Datei erledigte ProLogic-Dos hingegen mit Bravour. Hier zeigte sich das schnellste System von seiner besten Seite: 13 Sekunden war die beste Zeit zum Erstellen der Datei. Das Löschen dieser Datei dauerte zwar noch 9 Sekunden. Das Entfernen von 50 Testfiles mit dem SCRATCH-Befehl hingegen war schon nach 46 Sekunden passiert.

Bei unserer Testdiskette für VALIDATE zeigte sich das ProLogic-Dos ebenfalls von seiner besten Seite. Nach Dolphindos 2.0 – das einen Spezial-Algorithmus verwendet – lag ProLogic-Dos mit 125 Sekunden an zweiter Stelle.

Neben Dolphindos 2.0 zählt auch ProLogic-Dos zu den ausgereiften Systemen in unserem Test. Häufige Änderungen und neue Versionen sind bei ProLogic-Dos nicht zu befürchten, so daß der Kunde mit ziemlicher Sicherheit lange über ein aktuelles System verfügt.

Auch das Formatieren auf 40 Spuren ist unter ProLogic-Dos möglich. Für 35 Spuren werden beim Formatieren 20 Sekunden und für 40 Spuren werden 23 Sekunden benötigt. Der Speicherplatzgewinn ist dabei identisch mit dem bei Dolphindos 2.0 oder Professional Dos R4.0.

Die Funktionen, die ProLogic-Dos enthält, gleichen denen anderer Speeder. Neben ein paar zusätzlichen Befehlen im DOS zum bequemeren Arbeiten mit Files sind im Computer die Funktionstasten mit nützlichen Befehlen belegt.

Eigener Centronics-Anschluß

Auch ProLogic-Dos enthält eine Centronics-kompatible Schnittstelle. Diese wird allerdings nicht über den User-Port angeschlossen. Sie ist vielmehr am Expansion-Port als Platinenstecker herausgeführt.

Das hat natürlich Vorteile. Erstens bleibt der User-Port frei, zweitens ist dadurch der CIA 6526 vor Zerstörung geschützt, da er sehr empfindlich ist und auf Geräte am User-Port gerne »allergisch« reagiert, und drittens gibt es keine Probleme mit »Spezialkabeln«, da an der Centronics-kompatiblen Schnittstelle nur die Signale anliegen, die für den Centronics-Drucker benötigt werden.

Die bei Speedern üblichen Standardfunktionen sind auch bei ProLogic-Dos vorhanden. Dazu zählt eine Hardcopy vom Textbildschirm, eine erweiterte Reset-Routine zum Unterbinden von Autostart-Programmen, das Anhalten des Bildschirmscrolling und eine Reihe von Befehlen für die Floppystation, die sich auf die Erweiterungen beziehen. So zum Bei-

PROTEXT
TOP
ASS

ProLogic-Dos

Testaufgaben:		gemessene Zeiten:
LOAD (202 Blöcke)		5.5 Sekunden
SAVE (202 Blöcke)		10 Sekunden
SCRATCH (202 Blöcke)		7 Sekunden
SCRATCH (REL-Datei)		9 Sekunden
SCRATCH (50 Files à 1 Block)		46 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)		13 Sekunden
Diskette validieren		125 Sekunden
Diskette formatieren	(35 Spuren)	20 Sekunden
	(40 Spuren)	23 Sekunden

Technische Daten des Beschleunigungssystems:

Preis:		298 Mark
Computertyp:		C64, C128
Diskettenlaufwerk:		1541
Kapazität der Diskette (Spuren)		35/40 Spuren
	(Sektoren insgesamt)	664/749 Sektoren
Speicher in der Floppy		10 KByte
Anschluß des Parallelkabels		Expansion-Port
Taktfrequenz des Speeders		1/2 MHz
Speeder mit Schalter abschaltbar?		
	Floppy	ja
	Computer	ja

Wichtige zusätzliche Funktionen gegenüber dem Original-Betriebssystem:

- Centronics-kompatible Schnittstelle eingebaut (Expansion-Port)
- Directoryanzeige ohne Programmverlust
- Hardcopy von Textbildschirm
- OLD-Befehl
- Funktionstasten belegt
- Speeder in Stufen zurückschaltbar (für höhere Kompatibilität)
- Umschaltung der Taktfrequenz in der Floppystation
- Eingebauter Reset-Taster

Tabelle 5. ProLogic-Dos ist ein Renner unter den Floppy-Speedern

spiel Ein-/Ausschalten von VERIFY, Umschalten 35/40 Spuren oder Ein-/Ausschalten des Schnelladers, um eine höhere Kompatibilität zu geschützter Software zu erhalten.

ProLogic-Dos ist mit DIP-Schaltern vollständig abschaltbar, so daß jederzeit das Original-Betriebssystem in der 1541 und im C 64 wieder hergestellt werden kann. Auch der Anschluß eines zweiten Laufwerks mit der Gerätenummer 9 ist vorhanden, um mit zwei schnellen Laufwerken arbeiten zu können (zur Übersicht der Testdaten siehe Tabelle 5).

Das »Monstrum«

Als letzter Kandidat wartet noch ein System auf seinen Einsatz, das wir natürlich in unserem Vergleich bringen müssen, da es schon einen gewissen Marktanteil für sich verbuchen kann. Ob es sich aber dabei noch um einen Speeder der zweiten Generation handelt, ist sicher fraglich. Gemeint ist TurboTrans Plus und zwar die Version 3.0 (Bild 5). Bei diesem Beschleunigungssystem kann man ruhigen Gewissens schon von einem Speeder der dritten Generation mit eingebauter RAM-Disk sprechen, denn um etwas anderes handelt es sich hier nicht.

Das sagt einem auch schon der Preis, der zwar gemessen am Lieferumfang sicher nicht zu hoch, verglichen mit anderen Beschleunigungssystemen aber als enorm zu bezeichnen ist. TurboTrans Plus 3.0 kostet in der kleinsten Ausbaustufe 449 Mark. Es enthält dabei eine Platine für die Floppystation

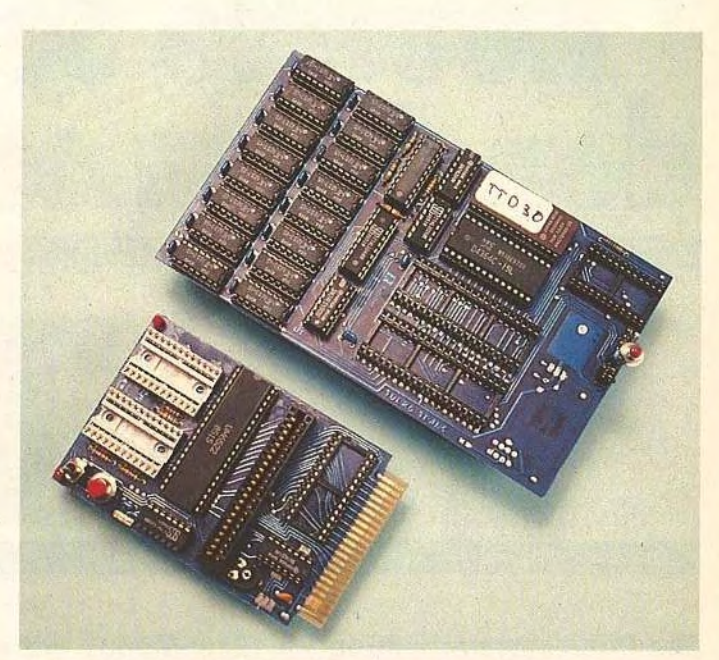


Bild 5. TurboTrans 3.0, Beschleuniger mit zwei RAM-Disks

mit 256 KByte dynamischen RAMs, eine Platine für den Expansion-Port des Computers mit Parallelkabel-Anschluß zur 1541 und einen Adaptersockel für das Kernel im Computer. Das neue Betriebssystem befindet sich auf der Platine im Expansion-Port. Zusätzlich bekommt der Käufer ein kleines Büchlein, das auf seinen 50 Seiten eine sehr ausführliche Einbau- und Bedienungsanleitung enthält. Sogar eine gründliche Systembeschreibung für die Programmierer ist vorhanden, so daß TurboTrans Plus auch leicht in die Entwicklung eigener Projekte eingebunden werden kann.

Für 99 Mark zusätzlich kann der Anwender auch die große Version von TurboTrans Plus 3.0 kaufen (512 KByte RAM). Hier können dann gegenüber zur »kleinen« Version gleich zwei komplette Disketteninhalte in die RAM-Disk eingelesen werden, so daß sehr bequem gearbeitet werden kann.

Zum Lieferumfang von TurboTrans Plus 3.0 gehört aber auch noch eine Diskette, die unter anderem ein schnelles Kopierprogramm und ein Codeschloß enthält. Mit dem Codeschloß können ganze Disketten vor unerlaubtem Zugriff gesichert werden.

Unsere Tests mußten wir jetzt natürlich auf zwei Stufen aufbauen. Zum einen kann TurboTrans Plus 3.0 auch betrieben werden, ohne eine Diskette ins RAM einzulesen. TurboTrans Plus 3.0 arbeitet dann wie ein »normaler« Speeder. Andererseits können sämtliche Funktionen auch in der RAM-Disk ausgeführt werden. Hier sind dann keinerlei »mechanische Verzögerungen« mehr vorhanden, so daß wir wahre Traumzeiten erwarten können.

Im Augenblick arbeitet TurboTrans Plus 3.0 noch unter 1 MHz Taktfrequenz. Das Betriebssystem ist aber softwaremäßig schon für den 2-MHz-Betrieb ausgelegt, so daß Sie, falls es Ihr Laufwerk verkraftet, durch eine kleine Bastelei (Einbau eines Umschalters bei IC UD 5) auf 2 MHz umstellen können. Laut Roßmöller ist aber ein TurboTrans in Vorbereitung, das immer auf 2 MHz laufen wird. Die bisherige Version läuft nämlich leider noch nicht mit jeder Floppy 1541 unter dieser hohen Taktfrequenz. Unsere Messungen wurden deshalb mit 1 MHz vorgenommen.

Das Laden von 202 Blöcken dauert unter TurboTrans Plus 3.0 genau 10 Sekunden. Befindet sich die Diskette im RAM, so schrumpft die Ladezeit auf 2,3 Sekunden. Beim Speichern zeigt sich TurboTrans Plus 3.0 erstaunlich behäbig. Hier müssen Sie ganze 101 Sekunden warten, bis das Programm auf der Diskette steht. Im RAM geht es zwar schneller, kommt

an unsere anderen Testkandidaten erstaunlicherweise aber nicht heran: 21 Sekunden für das Speichern im RAM.

Das Löschen von 202 Blöcken dauert auf der Diskette genau 25 Sekunden. Demgegenüber braucht das Löschen im RAM nur 2,5 Sekunden.

Die relative Datei wurde von TurboTrans Plus 3.0 in 56 Sekunden erstellt und in 13 Sekunden wieder gelöscht. Im RAM erforderte der gleiche Vorgang eine Wartezeit von 2 Sekunden, das Löschen nahm 1,3 Sekunden in Anspruch. Für die 50 Testfiles benötigte der SCRATCH-Befehl auf der Diskette 67 Sekunden. Im RAM war die ganze Angelegenheit nach 4 Sekunden vergessen.

Wie die Konkurrenz, so kann auch TurboTrans Plus 3.0 mehr als 35 Spuren formatieren. Hier haben sich die Entwickler sogar etwas ganz Besonderes einfallen lassen. Insgesamt kann TurboTrans Plus 3.0 mit 41 Spuren arbeiten. Das sind dann »766 BLOCKS FREE« oder anders ausgedrückt 191,5 KByte Speicherplatz auf der Diskette. Da jedoch nicht alle Laufwerke so viele Spuren bearbeiten können, erlaubt TurboTrans Plus 3.0 die Angabe einer beliebigen Spurnummer zwischen 35 und 41 als Maximum. Wollen Sie nur 39 Spuren formatieren, so erscheint die Meldung »732 BLOCKS FREE«, und Sie haben dann 183 KByte Speicherplatz zur Verfügung.

Das Formatieren dauert auf der Diskette zwischen 18 Sekunden (35 Spuren) und 21 Sekunden (41 Spuren). TurboTrans Plus 3.0 ist somit einer der Schnellformatierer unter den Speedern, obwohl während des Formatierens ein VERIFY erfolgt. Ohne VERIFY wären die 41 Spuren in nur 12 Sekunden formatiert.

Bei allen Zeitangaben für TurboTrans Plus 3.0 im RAM darf man allerdings nicht vergessen, daß die Diskette jeweils vor den Aktionen in das RAM eingelesen und hinterher wieder zurückgeschrieben werden muß, damit die Informationen nicht verlorengehen. Dieses Einlesen dauert im Augenblick noch 25 Sekunden (1 MHz). Das Zurückschreiben ist in 27 Sekunden passiert, wobei die eingelegte Diskette nicht formatiert zu sein braucht.

Komfortables Betriebssystem

An Zusatzfunktionen bietet TurboTrans Plus 3.0 dem Anwender eine ganze Menge. Es fehlen hier zwar die RS232-Routinen im Betriebssystem. Dafür bekommt der Anwender neben eingebautem Maschinensprache-Monitor, Centronics-kompatibler Schnittstelle, Hardcopy vom Textbildschirm und OLD-Befehl noch einen stark erweiterten Basic-Editor. Dieser enthält eine automatische Zeilennummernvorgabe, einen Bildschirm-Reset auf Tastendruck, zusätzliche Cursor-Funktionen auf Tastendruck und einiges mehr.

Dem bisherigen Schema (TurboAccess) folgend, sind auch bei TurboTrans Plus 3.0 keine Funktionstasten belegt. Alle zusätzlichen Funktionen und auch der eingebaute Monitor werden über Tastenkombinationen mit der <CTRL>-Taste aufgerufen.

Auch eine ganze Menge an zusätzlichen Befehlen für die Floppystation sind verfügbar. Das beginnt beim Ein- und Ausschalten von TurboTrans Plus 3.0 und geht bis hin zum Schützen von Files vor dem Löschen (eine Übersicht aller Testdaten zeigt Tabelle 6).

Ein gutes System also, das sich (leider), wie auch Professional Dos R4.0, noch in einer permanenten Weiterentwicklung befindet. Man sollte sich beim Kauf dieser Systeme vorher genauestens über das Entwicklungsstadium und die neueste Version informieren. Roßmöller gewährt dabei auch einen Update-Service. Es läßt sich aber nicht bestreiten, daß der zusätzliche Aufwand ein, wenn vielleicht auch kleines, Ärgernis für den Kunden darstellt.

Wir haben Ihnen mehrere aktuelle Floppy-Speeder vorgestellt. Jetzt wollen wir noch einmal Bilanz ziehen, welcher Speeder für welche Anwendungen geeignet ist. In Tabelle 7 sehen Sie die Testdaten des Original-Commodore-DOS zum Vergleich.

Das Ergebnis

Das beste Preis-/Leistungsverhältnis von allen Systemen besitzt zweifelsohne das Dolphindos 2.0. Mit einem Preis von 198 Mark liegt es sogar noch unter den Speedern der ersten Generation, wartet aber mit Leistungsdaten auf, die teilweise sogar den Hochgeschwindigkeitsfavoriten ProLogic-Dos in den Schatten stellen. Aber es geht nicht nur um die Geschwindigkeit. Entgegen sämtlicher Feilscherei um Zehntelsekunden bietet Dolphindos 2.0 ein durchdachtes und sehr anwenderfreundliches Betriebssystem, dessen Bedienung eine reine Freude ist. Zu erwähnen ist vor allem auch die Kompatibilität zu SpeedDos Plus, die dieses System auszeichnet. Wir konnten kein Programm finden, das unter SpeedDos Plus läuft, unter Dolphindos 2.0 hingegen nicht.

Haben Sie schon ein SpeedDos oder ein Floppy-Flash, so bietet sich der Einkauf des Professional Dos R4.0 an. Dieses System ändert grundlegende Details nur im Diskettenlaufwerk, während das Betriebssystem im Computer so kompatibel wie möglich gelassen wird. Ein sehr zuverlässiges und

TurboTrans Plus 3.0

Testaufgaben:		gemessene Zeiten:
		Disk/RAM
LOAD (202 Blöcke)		10/2.3 Sekunden
SAVE (202 Blöcke)		101/21 Sekunden
SCRATCH (202 Blöcke)		25/2.5 Sekunden
SCRATCH (REL-Datei)		13/1.3 Sekunden
SCRATCH (50 Files à 1 Block)		67/4 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)		56/2 Sekunden
Diskette validieren		150/10 Sekunden
Diskette formatieren	(35 Spuren)	18/0.1 Sekunden
	(40 Spuren)	20/0.2 Sekunden

Technische Daten des Beschleunigungssystems:

Preis:		449 Mark
Computertyp:		C 64, C 128
Diskettenlaufwerk:		1541
Kapazität der Diskette (Spuren)		35 bis 41 Spuren
	(Sektoren insgesamt)	664/766 Sektoren
Speicher in der Floppy		266 bis 522 KByte
Anschluß des Parallelkabels		Expansion-Port
Taktfrequenz des Speeders		1/2 MHz
Speeder mit Schalter abschaltbar?		
	Floppy	ja
	Computer	ja

Wichtige zusätzliche Funktionen gegenüber dem Original-Betriebssystem:

- Centronics-kompatible Schnittstelle eingebaut (User-Port)
- Directoryanzeige ohne Programmverlust
- Hardcopy von Textbildschirm
- OLD-Befehl
- Bildschirm-Reset auf Tastendruck
- eingebauter Maschinensprache-Monitor
- verschiedene Reset-Routinen zum Überspringen einer Modul-Kennung oder zum Sichern von Variablen und Basic-Programm
- Eingebauter Reset-Taster
- Speeder in Stufen zurückschaltbar (für höhere Kompatibilität)

Tabelle 6. Das Komfortsystem: TurboTrans Plus 3.0

Original-Dos

Testaufgaben:		gemessene Zeiten:
LOAD (202 Blöcke)		130 Sekunden
SAVE (202 Blöcke)		103 Sekunden
SCRATCH (202 Blöcke)		26 Sekunden
SCRATCH (REL-Datei)		16 Sekunden
SCRATCH (50 Files à 1 Block)		79 Sekunden
REL-Datei erstellen (200 Datensätze à 140 Zeichen)		60 Sekunden
Diskette validieren		230 Sekunden
Diskette formatieren	(35 Spuren)	85 Sekunden
	(40 Spuren)	-

Tabelle 7. Die Testwerte des Commodore Dos V2.6

sicheres System, dessen Vorteil es ist, daß sich der Anwender nicht großartig umgewöhnen muß.

Für die Geschwindigkeitsfanaktiker unter Ihnen bietet sich ProLogic-Dos an. Es erreichte im Test zwar keine einzelnen Traumwerte, dafür ist das Geschwindigkeitsniveau aber insgesamt am höchsten. Abschreckend ist vielleicht der relativ hohe Preis im Vergleich zu den anderen Kandidaten. Man sollte aber berücksichtigen, daß ProLogic-Dos aus sehr aufwendiger Hardware besteht, die von der Anschlußbelegung am Computer her sicherlich die beste Lösung darstellt.

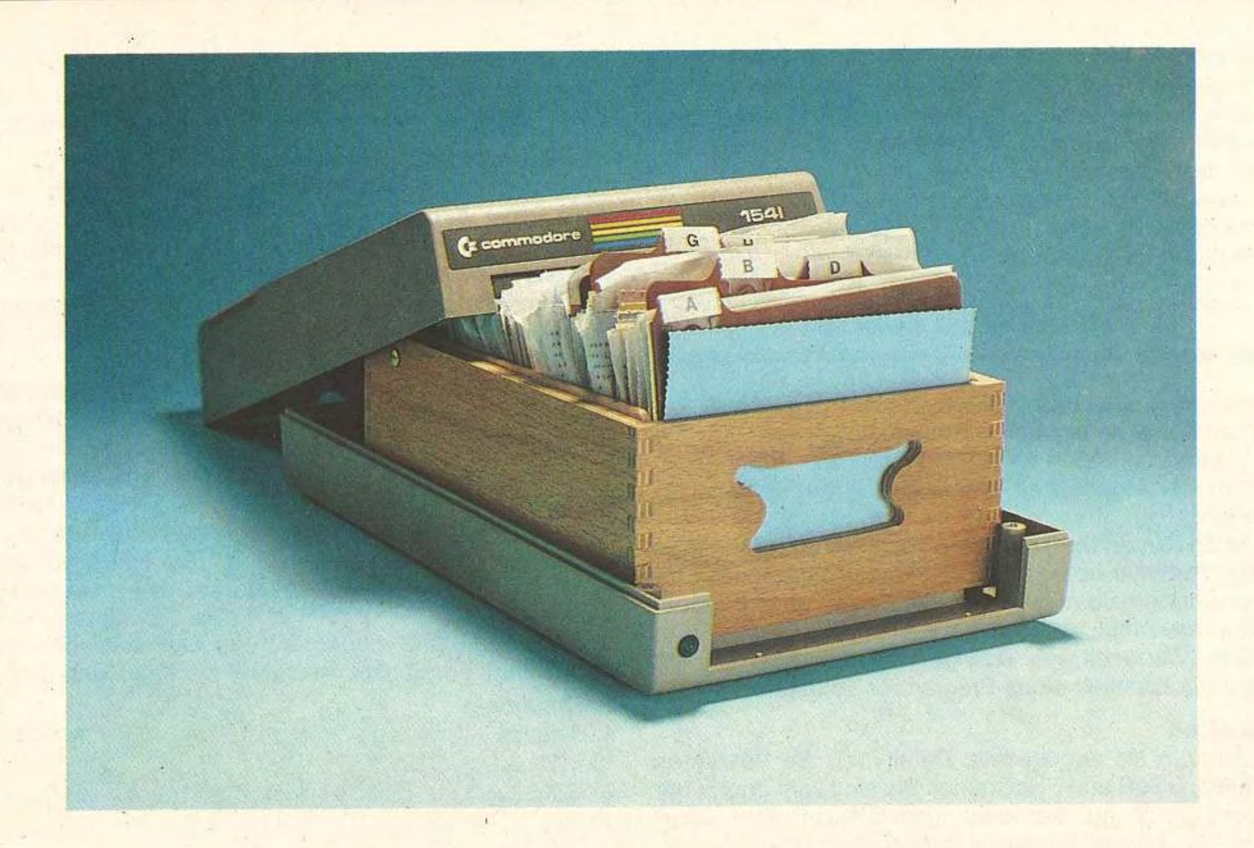
Haben Sie oft mit Compilern oder Assemblern zu tun oder kopieren Sie oft Disketten und haben nur ein Laufwerk? Dann dürfte Sie TurboTrans Plus 3.0 interessieren. Durch die eingebaute RAM-Disk bekommen Problemlösungen mit vielen Diskettenzugriffen auf einmal ganz andere Dimensionen. Sie warten nicht mehr Minuten, sondern nur noch ein paar Sekunden für Compiler- oder Assemblerdurchläufe und das ohne jede Belastung der Mechanik der 1541. Natürlich hat dieser Luxus auch seinen Preis. Können Sie TurboTrans Plus 3.0 jedoch professionell einsetzen, so ist der Preis gerechtfertigt.

Es klingt zwar ein wenig betrüblich, aber für die Speeder der ersten Generation scheint der Zug abgefahren zu sein. Spätestens seit es Dolphindos 2.0 zu einem günstigeren Preis gibt als beispielsweise SpeedDos Plus, ist es nicht mehr leicht einzusehen, warum man sich für einen der »Kleinen« entscheiden sollte. Hier leisten die neuen Produkte um einiges mehr.

Haben Sie einen C128 mit einer Floppy 1541, so bleibt Ihnen die Welt der Speeder trotzdem nicht verschlossen. Alle Speeder der zweiten Generation gibt es auch für den C128. Sie unterstützen dabei sowohl den C64-Modus als auch (siehe hierfür Tabellen 1 bis 6) den C128-Modus oder CP/M.

Die Kompatibilität aller Speeder zu geschützter Software ist sehr hoch. Hier stellen sich Dolphindos 2.0 und Professional Dos R4.0 an die Spitze. ProLogic-Dos bekommt teilweise Probleme mit Software, die keine 2 MHz in der Floppystation verträgt und TurboTrans muß meistens auf Diskettenbetrieb (ohne RAM-Disk) umgeschaltet werden. Da jedoch alle Speeder in Stufen zurückgeschaltet werden können bis hin zum Original-Betriebssystem, können selbst schwierigste Programme geladen werden. (ks)

SpeedDos Plus: Elektronik-Service Christoph Dichte, Fährstraße 33, 2212 Brunsbüttel
TurboAccess/TurboTrans 3.0: Roßmöller GmbH, Maxstraße 50-52, 5300 Bonn
Dolpindos 2.0: Jan Bubela, Engelsplatz 8, 6000 Frankfurt 63
Professional Dos R4.0: Mikrotronic System, Dipl. Ing. K. Roreger, Liebigstraße 28, 4780 Lippstadt
ProLogic-Dos: Jann Datentechnik, Glimmerweg 22, 1000 Berlin 47



Grundlagen der Dateiverwaltung

Die Verlockung, private Daten, wie zum Beispiel ein Telefon- oder Adressenverzeichnis mit dem C64 zu verwalten, ist groß. Wir erklären Ihnen leicht verständlich und mit vielen Beispielen, was Sie bei der Programmierung des Floppy-Laufwerks 1541 beachten müssen.

Zum Verständnis dieses Artikels müssen dringend ausreichende Basic-Kenntnisse vorausgesetzt werden. Andernfalls sei der Leser auf unser Sonderheft 7/86 (»Grundwissen«) verwiesen. Dort findet der C64-Einsteiger alle notwendigen Grundkenntnisse.

In diesem Artikel soll es zwar nur um Dateien des Floppy-Laufwerks 1541 gehen, vorweg jedoch trotzdem noch ein paar allgemeine Informationen zu Datasette, Harddisk und Floppy-Station.

Das Speichern von Daten auf Kassette ist, bei den Geschwindigkeiten der Commodore-Datasetten unrentabel. Weder ist die Datasette fähig, relativ zu adressieren (siehe später), noch ist sie schnell genug, um das Speichern in sequentieller Form lohnend zu machen. Der einzige Vorteil ist die Preislage, in der sich Kassettengeräte im Vergleich zu den Floppy-Laufwerken befinden. Wenn man aber die enormen Nachteile mit den Preisvorteilen vergleicht, so muß man feststellen, daß, zumal man heutzutage bei der richtigen Adresse Disketten zu einem Preis ab drei Mark erhalten kann, sich eine Datasette für jeden ernsteren oder intensiveren Computer-Anwender nicht lohnt. Das ist ein hartes Wort, aber alle Floppy-Besitzer, die schon einmal mit einer Datasette gearbeitet haben, werden mir zustimmen.

Wenn man nun die Harddisks (auch Festplatten-Laufwerke genannt), mit den Floppy-Disk-Laufwerken vergleicht, so wird man feststellen, daß die Harddisks im Vergleich zu der Speicherkapazität der Floppy-Disks (170 KByte) wahrhaft astronomische Größen von mehr als 20 MByte erreichen. Das klingt nun sehr schön, allerdings erreichen auch die Preise solch astronomische Höhen (meist um 2000 bis 3000 Mark). Vorteile einer Harddisk zeigen sich vor allem dann, wenn man sehr große Tabellen und »Daten-Monster« verwalten muß, da dank der parallelen Datenübermittlung und der höheren internen Zugriffszeiten beim Schreiben und Lesen die Geschwindigkeit der Datenübermittlung der der Floppy-Station 1541 weit überlegen ist.

Noch mal einige Worte zu denjenigen, die bis vor kurzem mit der Datasette gearbeitet haben: Zuerst zu den Sekundäradressen: Während es beim Kassettenlaufwerk deren drei gibt (0, 1 und 2 für Lesen, Schreiben und Schreiben mit »Band-Ende kennzeichnen«), gibt es beim Floppy-Laufwerk nur deren zwei (0 und 1) mit derselben Bedeutung wie bei der Kassette. Allerdings sind diese beiden Sekundäradressen bei der Floppy-Station von völlig untergeordneter Bedeutung. Bei Datenzugriff durch den Benutzer wird der Lese- oder Schreibmodus im Filenamen übermittelt. Außerdem kennt die Floppy-Station im Gegensatz zur Datasette mehrere Dateitypen, aber mehr darüber etwas später.

Zuerst möchte ich aufzählen, welche Dateitypen existieren, um nachher näher auf die einzelnen Typen einzugehen. Gleichzeitig werde ich auch Gebrauch und Anwendung der sequentiellen und relativen Dateien etwas näher erläutern.

Es gibt drei Arten von Dateitypen:
1. Die sequentiellen Dateiarten
2. Die relativen Dateien
3. Die Random-Access-Dateien.

1. Sequentielle Dateien

Man bezeichnet diejenigen Dateitypen als sequentiell, bei denen man nur in einer starren Reihenfolge auf die gespeicherten Daten zugreifen kann, und nicht wie bei anderen Dateitypen die Möglichkeit hat, auf das n-te Byte direkt zuzugreifen.

Der Vorteil dieses Dateityps liegt vor allem in der Einfachheit seines Aufbaus, sein größter Nachteil ist die lange Wartezeit, wenn innerhalb der Datei ein bestimmtes Element gesucht werden muß. Vor allem bei sehr großen Dateien kann eine solche Wartezeit sehr unangenehm sein oder gar der Benutzerfreundlichkeit eines Programms schaden.

1.1 Dateiarten

Es existieren als sequentielle Dateitypen die Dateiarten SEQuentiell, USeR und PRoGramm. Sie sind vom Zugriff her alle absolut gleich, nur die Kennung im OPEN-Befehl ändert sich, man benutzt meistens die Anfangsbuchstaben des Filetyps. Man muß darauf achten, daß nur jeweils drei sequentielle Dateien gleichzeitig geöffnet sein können.

Es muß noch mal gesagt werden: Diese drei Dateitypen sind vom internen Aufbau auf der Diskette her absolut gleich, nur im Directory werden sie als unterschiedlich angezeigt. Der Grund ist wohl vor allem, daß diese Files jeweils auch unterschiedlich genutzt werden; zum Beispiel die Programmdateien eben für Programme, sequentielle Dateien hingegen für Benutzer-Daten.

1.2 Zugriff auf sequentielle Dateien

Zuerst einmal die benötigten Basic-Befehle:
- OPEN Kanal, Gerät, Sekundäradresse, "Name, Filetyp (S,P oder U), Modus (R oder W)"
- CLOSE Kanal
- PRINT# Kanal, Text (;)
- INPUT# Kanal, Text
- GET# Kanal, Text

Kanal: Kennzahl für den C64, vom Benutzer frei wählbar (zwischen 1 und 127)

Gerät: Adresse des Peripheriegerätes (für die Floppy-Station meist 8 oder 9)

Sekundäradresse: Floppy-interne Kanal-Kennzahl (liegt für unsere Zwecke zwischen 1 und 14, muß ungleich 15 sein, da diese den Kommandokanal des Floppy-Laufwerks anspricht).

Name: Der Name des gewünschten Files

Filetyp: Kennbuchstabe, der entscheidet, welcher Dateityp geöffnet werden soll (SEQ, PRG oder USR)

Modus: Ist meistens »R« oder »W«, seltener »A«, für Notfälle »M«.

»W« bedeutet, daß die betreffende Datei zum Schreiben geöffnet werden soll. Wenn sie schon existiert, wird ein »FILE EXISTS ERROR« gemeldet.

»A« erlaubt es, eine sequentielle Datei zum Schreiben zu öffnen. Dabei werden die zu schreibenden Daten an ein schon existierendes File angehängt.

»R« mit diesem Buchstaben eröffnet man eine schon existierende Datei, um sie zu lesen.

»M« Wenn eine Datei nicht ordnungsgemäß geschlossen wurde, so steht im Directory hinter dem Filenamen ein Stern, und diese Datei kann nicht mehr gelesen werden. Um jetzt wenigstens einen Teil der Daten zu retten, muß man das File im Modus M eröffnen und die Daten in ein neues File ablegen.

1.3 Besonderheiten von PRINT# und INPUT#

PRINT#: Bei diesem Befehl darf man nicht vergessen, daß nach jeder Ausgabe, wie auf dem Bildschirm auch, ein CHR$(13), das heißt ein »Return« gesendet wird. Um dies zu vermeiden (wenn man zum Beispiel auf der Diskette Strings zusammensetzen will), muß man nach dem Print-Befehl ein Semikolon setzen, das das Senden eines »Returns« unterdrückt.

INPUT#: Hier müßte man Commodore eigentlich Vorwürfe machen, daß sie einen so nützlichen Befehl so schlecht ausgebaut haben. Er zeigt einige Schwächen, die die Nutzung von Dateien erschweren, und vor allem dem Anfänger Probleme bereiten können. Die eine Schwäche ist die Beschränkung der Länge der ausgelesenen Strings auf etwas mehr als 80 Zeichen, die andere ist die »EXTRA IGNORED«-Fehlermeldung bei bestimmten Satzzeichen. Das Unangenehmste freilich ist, daß der INPUT#-Befehl Stringteile zwischen einem Komma und einem »Return« entweder als selbständige Strings betrachtet, oder aber sie völlig ignoriert!

1.4 Beispiele

1. Beispiel

```
10 OPEN 2,8,2,"TEST,S,W"
20 OPEN 1,8,15 : INPUT#1,A,B$,C,D : IF A = 0 THEN 40
30 PRINT "FEHLER :" : PRINT A;B$;C;D : CLOSE 2 :
   CLOSE 1 : END
40 PRINT#2,"DIES IST EIN GANZER SATZ!"
50 CLOSE 2
60 INPUT#1,A,B$,C,D : IF A <> 0 THEN 30
70 PRINT"OK" : CLOSE 2 : CLOSE 1 : END
```

Erklärung: In Zeile 10 wird auf dem Kanal 2 ein sequentielles File zum Schreiben neu eröffnet.

Dann wird in Zeile 20 der Kanal 1 mit der Sekundäradresse 15 geöffnet. Dies ist der Kommandokanal des Floppy-Laufwerks. Hier wird in der gleichen Zeile auch die Status- oder Fehlermeldung der Floppy-Station ausgelesen. In der Variablen A ist die Nummer der Fehlermeldung gespeichert, 0 bedeutet OK.

Wenn A ungleich 0 ist, also ein Fehler vorliegt, wird in Zeile 30 die Fehlermeldung ausgegeben und das Programm abgeschlossen.

In Zeile 40 wird dann der Text auf Kanal 2 ausgegeben, also in die Datei geschrieben.

Dann wird in Zeile 50 das File geschlossen. Wenn man dies nicht machen, sondern einfach die Diskette aus dem Laufwerk entfernen würde, wäre das File im Directory als fehlerhaft gekennzeichnet und nicht mehr zu gebrauchen.

In Zeile 60 wird wieder der Floppystatus kontrolliert und bei aufgetretenem Fehler in der Zeile 30 ausgegeben.

Zeile 70 schließt einfach den Kommando-Kanal und beendet dieses kleine Programm.

2. Beispiel:

```
10 OPEN 1,8,1, "TEST,S,R"
20 INPUT#1,A$
30 PRINT A$
40 CLOSE 1 : END
```

Erklärung: Dieses kleine Programm öffnet unsere sequentielle Datei »TEST« zum Lesen.

In Zeile 10 wird die Datei geöffnet und der Kanalnummer 1 zugewiesen.

Dann wird aus dem Kanal 1 und somit aus unserer Datei ein String gelesen und auf dem Bildschirm ausgegeben.

Anschließend wird Kanal 1 wieder geschlossen. In diesem Fall, beim Lesen einer Datei, ist es eigentlich ungefährlich, die Diskette aus dem Laufwerk zu nehmen, wenn der Kanal noch offen ist (rote LED an der Floppy-Station leuchtet), man sollte

aber trotzdem konsequent vorgehen und nicht benötigte Dateien immer mit einem CLOSE schließen.

3. Beispiel:

```
10 OPEN 2,8,2, "TEST,S,A"
20 PRINT#2, "DAS IST EIN ZWEITER SATZ"
30 CLOSE 2 : END
```

Erklärung: Dieses Mini-Programm öffnet das File »TEST« im »Append«-Modus. Das heißt, die vorhandenen Einträge in der Datei bleiben erhalten, und die neuen Einträge werden hinten angehängt.

Wenn man jetzt diese Datei wieder auslesen will, dann kann man so vorgehen wie im Beispiel 2, nur müssen jetzt statt einem INPUT#-Befehl zwei ausgeführt werden.

4. Beispiel:

```
10 OPEN 1,8,1, "LISTING-TEST,U,W"
20 CMD1:LIST
CLOSE 1
```

Erklärung: Wenn man dieses Programm laufen läßt, »listet« es sich selbst auf Diskette, und kann mit Hilfe des nächsten Beispiels ausgedruckt werden.

In Zeile 10 wird ein User-File zum Schreiben geöffnet, dann werden in Zeile 20 sämtliche Ausgaben des Computers (also auch der LIST-Befehl) zu Kanal 1 umgeleitet und somit auf unsere Datei ausgegeben. Anschließend muß der Kanal 1 im Direktmodus geschlossen werden, da das Programm nach einem LIST automatisch abgebrochen wird.

5. Beispiel:

```
10 OPEN 2,8,2, "LISTING-TEST,U,R" : OPEN 4,4
20 GET#2,A$:B=ST:PRINT#4,A$;
30 IF B AND 2↑6 <> 0 THEN 20
40 CLOSE 2 : CLOSE 4 : END
```

Erklärung: Dieses Programm gibt ein Basic-Programm, das auf Diskette mittels CMD gelistet wurde, auf einem Drucker aus.

Zuerst wird in Zeile 10 das File auf Diskette geöffnet, dann der Kanal zum Drucker (Geräteadresse 4).

Jetzt wird in Zeile 20 ein Zeichen aus der Datei geholt, dann der Status in eine andere Variable gerettet und das eine Zeichen ohne »Return« (deshalb das Semikolon nach dem PRINT-Befehl) auf dem Drucker ausgegeben.

Dann wird der gerettete Status auf Bit 6 kontrolliert, und, wenn dieses nicht gesetzt ist, das nächste Zeichen von Diskette geholt. Ist Bit 6 gesetzt, ist das Ende der Datei erreicht, und die Kanäle werden wieder geschlossen.

Noch eine kleine Erklärung zum Status in der Variablen ST: In diesem Status-Byte wird das sechste Bit auf eins gesetzt, wenn das letzte Byte des Files gelesen wurde. Bezeichnenderweise heißt dieses sechste Bit auch EOF (=End of File). Allerdings darf der Anwender nicht vergessen, das Status-Byte sofort nach dem GET oder gegebenenfalls nach einem INPUT in einer Variablen (hier: B) zu retten, da sich der Wert dieses Byte nach einem PRINT-Befehl an die Peripherie (hier: Drucker) auch ändern kann.

2. Relative Dateien

Relative Dateien sind nun etwas ganz anderes, als die vorher behandelten sequentiellen Dateien. Von Ihnen gibt es nur einen Typ – eben relativ, im Gegensatz zu den sequentiellen, wo es eine feinere Unterteilung in einzelne Arten gab.

Da relative Dateien vom C 64-Basic nicht unterstützt werden, müssen alle Kommandos relativ umständlich erfolgen. Die aus dieser Verwaltungsart resultierenden Vorteile überwiegen die vorhandenen Nachteile aber bei weitem.

Man kann eine relative Datei am besten mit einem eindimensionalen Stringfeld vergleichen, wo alle Strings die gleiche Länge haben. So wie man im Speicher diese einzelnen Array-Teile adressieren kann, ist es auch in relativen Dateien möglich, einen bestimmten Datenblock (=Record) zu lesen oder zu schreiben. Das erklärt natürlich, warum diese Dateien als »schnell« bezeichnet werden: Auf einen einzelnen Record, auch wenn er ganz am Schluß der Datei ist, kann sofort zugegriffen werden, da die Floppy-Station intern noch eine Tabelle verwaltet, auf der die genaue Position der Recordanfänge verzeichnet ist.

Noch ein kleiner Tip: Man sollte sich mit diesem Dateityp wirklich nur auseinandersetzen, wenn man unbedingt kurze Zugriffszeiten benötigt und schon etwas Erfahrung im Umgang mit dem Floppy-Laufwerk hat, da, wie gesagt, die Anwendung dieser Dateien doch etwas kompliziert werden kann.

2.1 Zugriff auf relative Dateien

Zum Öffnen einer relativen Datei benutzt man folgenden Befehl:

```
OPEN 2,8,2, "NAME,L,"+CHR$(40)
```

Der OPEN-Befehl ist dem der sequentiellen Dateien fast gleich, man darf aber nicht vergessen, daß gleichzeitig immer nur eine relative Datei offen sein kann. Man kann allerdings dann noch eine sequentielle Datei öffnen, was aber später bei der Index-sequentiellen Dateiverwaltung besprochen werden soll.

Zurück zu unserem OPEN: Mit dem L nach dem Namen wird der Floppy mitgeteilt, daß eine relative Datei geöffnet werden soll. Die CHR$-Sequenz teilt dem Laufwerk mit, wie groß ein einzelner Record, ein einzelner Datenblock werden soll (in unserem Beispiel 40 Zeichen lang). Allerdings muß man beachten, daß nach dem ersten Öffnen, also nach dem Initialisieren der Datei, dieser Wert immer gleich bleiben muß. Denn wenn man beim Initialisieren die Datenblöcke für 40 Byte vorbereitet, und dann bei einem späteren Öffnen der Datei mitteilt, daß die Blöcke 80 Byte lang sein sollen, so wird sich das Laufwerk den Prozessor darüber zerbrechen, wie sie 80 Byte in 40 Stellen unterbringen soll. Wenn Sie die Datei zu einem späteren Zeitpunkt wieder benutzen wollen, werden Sie sich wundern, wie es in den einzelnen Records aussieht!

Nach erstmaligem Öffnen der Datei muß diese noch eingerichtet werden, das heißt, Sie müssen der Floppy-Station mitteilen, wieviele Records Sie bis auf weiteres benutzen wollen. Dazu müssen Sie den letzten Datensatz mit einem CHR$(255) markieren, worauf das Laufwerk alle noch nicht benutzten Datensätze selber einrichtet. Das kann eine Weile dauern, darum sollte man dieses Einrichten gleich am Anfang erledigen. Wenn später die Datei erweitert werden soll, muß man einfach den neuen letzten Datensatz einrichten. Nehmen wir an, Sie wollen 300 Records der Länge 40 freigeben:

```
10 OPEN 2,8,3,"NAME,L,"+CHR$(40)
20 OPEN 1,8,15
30 PRINT#1,"P"+CHR$(3)+CHR$(300 AND255)+CHR$(300/
   256)+CHR$(1)
40 PRINT#2,CHR$(255)
50 INPUT#1,A:CLOSE 1: CLOSE 2 : END
```

In Zeile 10 wird, wie schon besprochen, die Datei eröffnet und in Zeile 20 der Kommandokanal, den man zum Positionieren des Records benötigt.

Die Syntax des Positionierbefehls (Zeile 30), den man auf dem Kommandokanal senden muß, lautet wie folgt:

```
PRINT#Kanal,"P"+CHR$(Sekundäradresse)+CHR$(Lo)+
CHR$(Hi)+CHR$(Stelle)
```

»P«	Kennzeichen des Positionierbefehls
Sekundäradresse:	Sekundäradresse der relativen Datei im Open-Befehl
Lo:	niederwertiges Byte der Datensatznummer
Hi:	höherwertiges Byte der Datensatznummer

Stelle: Die Nummer des Zeichens innerhalb eines Records (Datensatz), auf das positioniert werden soll.

Somit wird in Zeile 30 auf das erste Zeichen des Record Nummer 300 positioniert. Die Aufspaltung der Recordadresse in Lo und Hi ist nötig, da mit einem CHR$-Befehl nur Werte zwischen 0 und 255 übergeben werden können, aber durchaus Datensatzadressen der Größe 500 oder 1000 nötig werden können!

In Zeile 40 wird dann ein CHR$(255) auf dem eigentlichen Dateikanal ausgegeben, was die Initialisierung startet.

Zum Schluß wird dann die eventuell blinkende Fehler-LED gelöscht, und beide Kanäle geschlossen.

Dieses Blinken sollte Sie beim Einrichten einer relativen Datei nicht irritieren, es tritt immer dann auf, wenn ein noch nicht existierender Record angesprochen werden soll. Falls die LED beim Einrichten einer größeren Datei überhaupt nicht leuchtet, so sollte Sie das nicht weiter stören, das ist normal.

2.2 Beispiele

1. Beispiel:

```
10 OPEN 2,8,3,"NAME,L,"+CHR$(40)
20 OPEN 1,8,15
30 PRINT#1,"P"+CHR$(3)+CHR$(1)+CHR$(0)+CHR$(1)
40 PRINT#2,"DAS IST JETZT WIRKLICH EIN LANGER
   TEXT! ";
50 CLOSE 1 : CLOSE 2 : END
```

Erklärung: Bei diesem Programm muß eigentlich nur erwähnt werden, daß in Zeile 40 beim PRINT#2 ein Semikolon steht. Wie man oben beim OPEN-Befehl sieht, ist ein Datensatz 40 Zeichen lang, aber mit »Return« wäre der Text in Zeile 40 ganze 41 Zeichen lang!

Wenn man jetzt diesen Text wieder lesen möchte, so muß man notgedrungen mit einer GET-Schleife arbeiten, da bei einem INPUT# bekanntlich ein »Return« gesucht wird.

Wenn man mit INPUT arbeiten möchte, was trotz allem doch noch einige Vorteile bietet, muß man immer darauf achten, daß kein »PRINT« vorkommt und im Record immer genügend Platz für ein »Return« bleibt.

2. Beispiel:

```
10 OPEN 2,8,3, "NAME,L,"+CHR$(40)
20 OPEN 1,8,15
30 PRINT#1,"P"+CHR$(3)+CHR$(1)+CHR$(0)+CHR$(1)
40 GOSUB 100:PRINT A$
50 CLOSE 1: CLOSE 2 : END
99 :
100 A$="":FOR I = 1 TO 40
110 GET#2,B$:IF B$ =CHR$(255) THEN 130
120 A$=A$+B$:NEXT
130 RETURN
```

Erklärung: Dieses Programm simuliert eine Art INPUT# in der Subroutine ab Zeile 100. Dort wird die maximale Anzahl zu holender Zeichen in der FOR-NEXT-Schleife festgelegt (gleich der Recordlänge), und dann so lange ein Zeichen nach dem anderen aus dem Record geholt, bis das nächste Zeichen ein CHR$(255) ist: das Leerkennzeichen.

Wenn man einmal einen String von nur 30 Zeichen Länge in einem 40-Zeichen-Record mit einem PRINT#; ablegt, so haben die restlichen zehn Zeichen den Wert 255, was in dieser GET-Routine also statt einem CHR$(13) als Endekennzeichen verwendet wird.

2.3 Index-sequentielle Dateien

Dieser umständliche Name beschreibt etwas eigentlich gar nicht so kompliziertes: Wie schon an verschiedenen Stellen angedeutet, kann man bei der VC 1541 nur eine sehr beschränkte Anzahl Dateien gleichzeitig öffnen; nämlich drei sequentielle Dateien oder eine relative und eine sequentielle Datei.

Diese Möglichkeit, gleichzeitig eine relative und eine sequentielle Datei zu öffnen, ist nun die Voraussetzung zur Index-sequentiellen Datenverwaltung.

Nehmen wir einmal an, Sie wollen ein persönliches Adreßregister mit einem nicht zu kurzen Kommentar zu jeder einzelnen Adresse oder Person aufbauen: Sie können jetzt eine Adresse nach Namen, Wohnort oder Arbeitsort alphabetisch ordnen. Eine sehr aufwendige Methode wäre jetzt, drei komplette relative Dateien aufzubauen, eben eine nach Namen, die nächste nach Wohnort etc. alphabetisch geordnet. Darum werden Sie wohl eher nur eine relative Datei, nach Namen geordnet, aufbauen, dazu aber zwei sequentielle Dateien, in denen nur entweder die Wohnorte oder die Arbeitsorte mit den entsprechenden Record-Nummern des eigentlichen Eintrags aufgeführt sind.

Wenn man jetzt nach eingegebenen Namen suchen will, so tut man das direkt in der relativen Datei und hat somit sofort den richtigen Eintrag. Wenn man aber nach eingegebenem Wohnort sucht, sucht man diesen in der sequentiellen Datei, wo der Eintrag vielleicht so aussieht: »UNTERLUNKHOFEN ... 0753«. Wie man sieht, gibt es hier maximal 9999 Records und die maximale Länge des Ortes sind 16 Buchstaben. Man sucht also in der Wohnortdatei »Unterlunkhofen«, liest die entsprechende Recordnummer (hier: 753) und findet so den entsprechenden Eintrag in der relativen Datei.

Natürlich muß man nach jeder Änderung der relativen Datei, da diese ja auch neu sortiert werden muß, die entsprechenden sequentiellen Dateien neu aufbauen.

Anmerkung: Es wird wohl besser sein, wenn man auch die Namen der relativen Einträge in einer sequentiellen Datei sortiert hält, da das Umordnen innerhalb der relativen Datei mit den Kommentaren doch sehr mühsam werden kann!

3. Random-Access-Dateien

Dieser Dateityp ist weder relativ noch sequentiell, oder besser gesagt, es kommt ganz darauf an, wie Sie ihn organisieren. Denn Random-Access bedeutet freier Zugriff. Mit anderen Worten: Der Benutzer kann diese Art von Dateien verwalten, wie er will. Es ist nun nicht schwer zu verstehen, daß bei einer so flexiblen Dateienverwaltung das Lesen und Schreiben von Daten sehr umständlich werden kann. Darum wird in diesem Artikel auch nur das Grundsätzliche erläutert.

Nun zur Datenstruktur: Die Daten werden in diesem Fall, anstatt wie bisher in Dateien, direkt in die einzelnen Blöcke auf der Diskette geschrieben. Man muß sich also immer selber merken, wo jetzt welche Daten stehen. Das macht man am besten, indem man auf einem festdefinierten Block eine Tabelle der Adressen der einzelnen Datensätze unterhält. (Oder man legt eine sequentielle Datei an, in der dann diese Tabelle gespeichert ist.) Diese Struktur imitiert also in gewisser Weise die relative Dateiverwaltung der Floppy-Station, ist aber etwas schwerer zu handhaben.

Man kann auch die sequentielle Datenstruktur imitieren, indem man am Anfang oder Ende eines jeden Datenblocks die Adresse des nächsten angibt.

3.1 Zugriff auf Random-Access-Dateien

Zuerst eine Liste sämtlicher neuer Befehle:

```
PRINT#1, "U1 3 0 X Y"
PRINT#1, "U2 3 0";X;Y
PRINT#1, "B-P 3 X"
PRINT#1, "B-A X Y"
PRINT#1, "B-F X Y"
OPEN 2,8,3,"#"
```

Man sieht, ganz normales Basic, aber die Anweisungen zwischen den Anführungszeichen sind für den Kommandokanal des Laufwerks bestimmt, und bewirken dort etwas ganz

bestimmtes. (Kommandokanal zum Beispiel =OPEN1,8,15.)
`U1 3 0 x y`: liest den Block (x,y) in den Puffer, der zur Sekundäradresse 3 des Laufwerks 0 gehört.
`U2 3 0 x y`: schreibt den Block (x,y) aus dem Puffer zur Adresse 3 auf die Diskette.
`B-P 3 x`: richtet den Zeiger des Puffers mit der Adresse 3 auf das x-te Byte.
`B-A x y`: sperrt den Block (x,y) in der BAM, der Block kann vor einem »Validate« (PRINT#1, "V") oder einem B-F durch reguläre Dateien oder Programme nicht mehr überschrieben werden.
`B-F x y`: gibt den Block (x,y) wieder frei.
`OPEN 2,8,3,»#«`: öffnet den Kanal 2 mit der Sekundäradresse drei als Datenpuffer.

Eine Empfehlung: Im Prinzip gibt es statt U1,U2 die Befehle B-R, B-W, allerdings haben sie gewisse Tücken, die sie für den normalen Benutzer sehr unattraktiv machen.

Vielleicht noch kurz ein Wort zur Blockstruktur auf Diskette:

Spurnummer:	Anzahl Sektoren:
01-17	21 (00-20)
18-24	19 (00-18)
25-30	18 (00-17)
31-25	17 (00-16)

Warnung: Die Spur 18 ist reserviert, hier befindet sich das Directory und die BAM!

Alle diese Befehle und entsprechenden Fehlermeldungen sind im gewöhnlichen Floppy-Handbuch oder im Floppy-Buch von Markt&Technik besser und ausführlicher erklärt, es würde aber den Rahmen des Artikels sprengen, wollte ich hier alles aufführen.

3.2 Beispiele

Schreiben eines Textes in den Block (1,0) und Sperren des Blocks:

```
10 OPEN 1,8,15 "I":OPEN 2,8,3,"#"
20 PRINT#1, "B-P 3 0"
30 PRINT#2, "HALLO DU DA!"
40 PRINT#1, "U2 3 0 1 0":REM oder:,"U2";3;0;A;B,
   wenn A und B die Werte 1 und 0 haben.
50 INPUT#1,A,B$,C,D:IF A <> 0 THEN PRINT A,B$,
   C,D:CLOSE 1: CLOSE 2 : END
60 PRINT#1,"B-A 1 0"
70 CLOSE 2:CLOSE 1:END
```

Lesen dieses Textes und Freigeben des Blocks:

```
10 OPEN 1,8,15,"I":OPEN 2,8,3,"#"
20 PRINT#1, "U1 3 0 1 0"
30 INPUT#1,A,B$,C,D:IF A<>OTHENPRINT A,B$,C,D:
   CLOSE1:CLOSE2:END
40 PRINT#1, "B-P 3 0"
50 PRINT#1, T$:PRINT T$
60 PRINT#1, "B-F 1 0"
70 CLOSE 2:CLOSE 1:END
```

3.3 Hash-Code-Dateien

Dieser Dateityp beruht auf dem Random-Access-Prinzip und hat noch einen Schuß relativer Datenverwaltung in sich. Wie Sie sicher wissen, gibt es auf einer Diskette 664 frei benutzbare Blöcke, sofern keine Daten auf Diskette gespeichert sind.

Wenn man nun wieder eine Adressenverwaltung aufbauen will und man annimmt, daß keine zwei Einträge gleich lauten, kann man sich vorstellen, daß zum Beispiel die Summe des ganzen Textes (ASCII-Code) direkt die Adresse eines Eintrages darstellt. Es ist klar, daß sich so sehr kurze Suchzeiten (praktisch gleich Null) ergeben, andererseits ist auch nur eine beschränkte Anzahl Einträge verwaltbar.

Noch einmal das Vorgehen: Nehmen wir an, jemand sucht den Eintrag »OTTO«. Die ASC-Summe dieser vier Buchstaben beträgt 326. Die Nummer des Blocks wäre dann (16,11), da 326/21 = 15 Rest 11 ist. Der tatsächliche Block ist nicht (15,11), da es ja keine Spur Null gibt. Man muß also zum Resultat noch 1 dazuzählen. Es wäre auch denkbar, das Zahlenergebnis als Record-Nummer einer relativen Datei einzusetzen.

Dies war das grobe Prinzip der Hash-Datenverwaltung, es bleibt nun dem Leser selbst überlassen, das Konzept zu verfeinern, indem er eine »narrensichere« Adressierungsmethode findet (bei dem vorhergenannten Beispiel gäben die Kombinationen »TOOT«, »STOP« oder »POST« auch den Wert 326!) und zum Beispiel mehrere Datensätze in einem Block erlaubt, indem er nicht 664, sondern 1328 verschiedene »Codes« erkennt und jeweils zwei (gerade-ungerade) im selben Sektor unterbringt.

Mit diesem Artikel hoffe ich, dem Leser das Floppy-Laufwerk um einiges schmackhafter gemacht zu haben, und auch einen kleinen Einblick in die »Welt der Datenverarbeitung« gegeben zu haben.

Noch eine kleine Bemerkung, eine Besonderheit der relativen Dateien auf dem Floppy-Laufwerk 1541: Achten Sie unbedingt darauf, den Record Nummer Null nie zu benutzen. Als ich ihn einmal verwenden wollte, fand ich Teile der Daten dieses Records in den Records eins bis vier, was sich doch etwas störend auswirken kann!

Außerdem sollten Sie, aus ähnlichen Gründen, nie auf das Byte Null eines Records positionieren und dann schreiben oder lesen wollen. (Germano Caronni/tr)

In die Geheimnisse der Floppy eingetaucht

Die Diskettenlaufwerke 1541, 1570 und 1571 sind Renner unter den Massenspeichern. Doch mit der passenden Literatur hapert es. Deshalb beschränken sich die meisten Anwender auf das Speichern und Laden von Programmen. Mit diesem Kurs lernen Sie, Ihr Laufwerk effektiver einzusetzen und es sogar zu manipulieren.

Daß die 1541 (die Angaben gelten auch für die 1570/1571 im 1541-Modus) ein sehr wandelbares Gerät ist, werden die meisten Benutzer wohl wissen oder zumindest erahnen. Man denke ja nur an den »Kleinkrieg« zwischen Softwareherstellern und Softwarepiraten, die sich gegenseitig das Leben schwer machen. Die meisten »Schlachten« liefert man sich hier im Inneren der Floppy-Station, die viel raffiniertere Methoden des Programmschutzes anbietet als der Commodore 64 selbst.

Aber auch Programme wie Hypra-Load beweisen die Flexibilität der Diskettenstation. Doch wie bei so vielen Dingen in der Commodore-Welt sind auch hier die Informationen rar, beziehungsweise in den Handbüchern gar nicht vorhanden. So wollen wir uns mit Ihnen an das Floppy-Laufwerk heran- und vorsichtig hineintasten. Angefangen bei grundlegenden Informationen über den Diskettenaufbau und den Befehlssatz des Laufwerks werden wir Schritt für Schritt in dessen Möglichkeiten zur Programmierung und Manipulation hinabtauchen. Was wird benötigt?

Nun, außer einem C 64 (C 128) und einer 1541/70/71, »nur« Basic-Erfahrungen, grundlegende Kenntnisse in Maschinensprache und ein wenig Geduld.

Bevor wir jedoch mit unserer ersten Tauchfahrt beginnen, tippen Sie bitte das beigefügte Programm EDDI (Listing 1) ein, sofern Sie nicht über einen eigenen Disk-Monitor verfügen. Auf die Bedienung von EDDI wird im einzelnen noch eingegangen.

Sehen wir uns jetzt erst einmal eine Diskette an. Die folgenden Erläuterungen beziehen sich auf eine formatierte Diskette.

Aufbau einer Diskette

Eine Diskette ist in 35 konzentrische Spuren (englisch: Tracks) aufgeteilt. Jede dieser Spuren enthält wiederum eine bestimmte Anzahl von Sektoren, die von außen nach innen abnimmt. Die genauen Zahlenverhältnisse stehen in Tabelle 1.

Die Spuren sind, beginnend mit der äußeren Spur, von 1 bis 35 durchnumeriert. Die Sektoren sind auf den Spuren in numerischer Reihenfolge gegen den Uhrzeigersinn angeordnet. Jeder Sektor enthält einen Block, das sind 256 Byte, an Information. Es kann jeder der 683 Blöcke auf der Diskette durch Angabe der jeweiligen Spur- und Sektornummer aufgerufen werden. Allerdings stehen davon dem Benutzer normalerweise nur 664 (1328) Blocks zur Verfügung, da das Betriebssystem der Floppystation die Spur 18 (18 und 53) für sich beschlagnahmt (für die 1571 gilt natürlich jeweils die doppelte Kapazität im C 128-Modus).

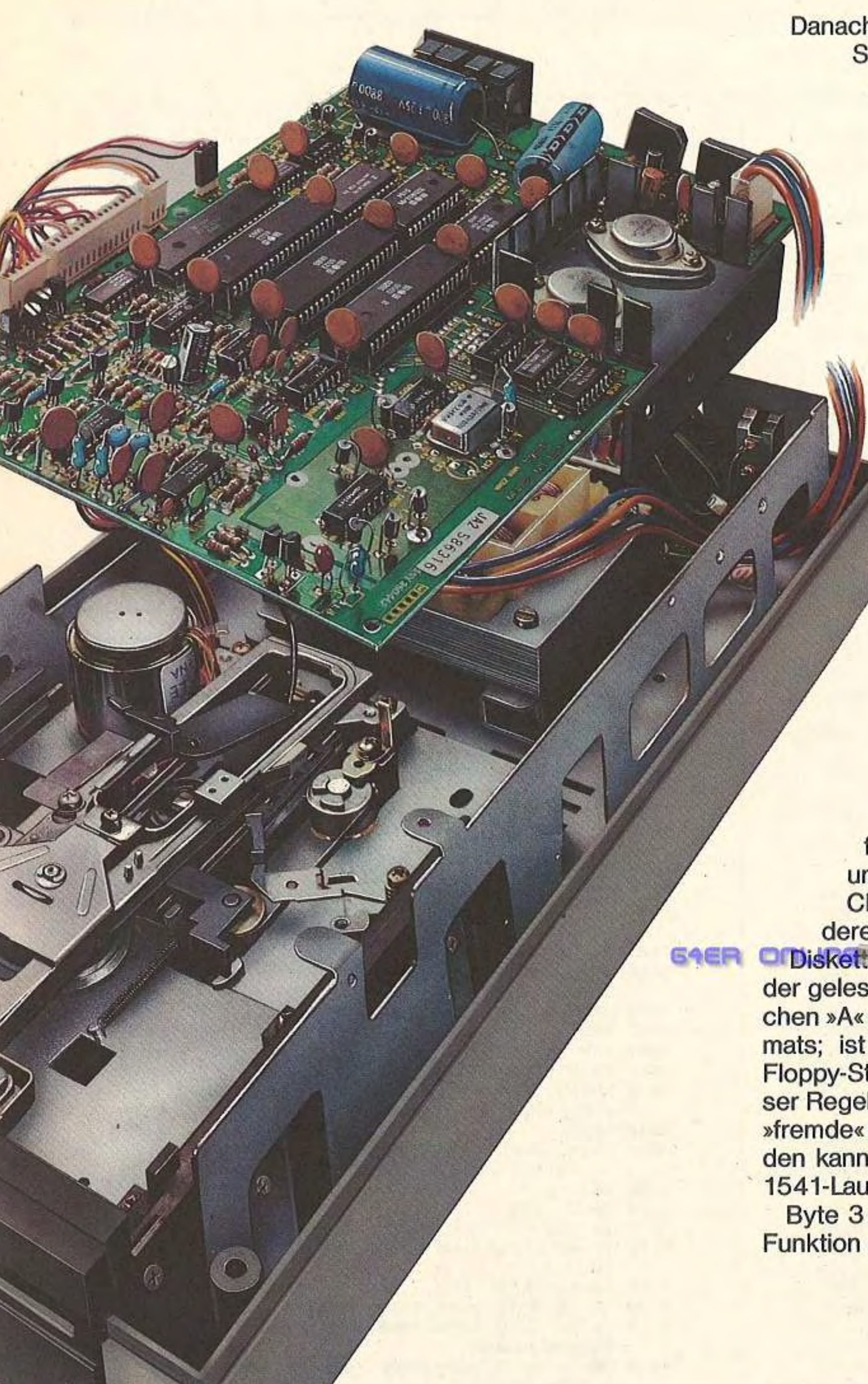
Für die nun folgenden Versuche wäre es sinnvoll, eine Diskette neu zu formatieren, mit der wir ein bißchen »spielen« können. Sehen wir uns nun erst einmal das Directory an (LOAD "$",8):

In der ersten Zeile stehen die Drive-Nummer (hier immer 0) und der Name der Diskette, sowie die ID und das Formatkennzeichen (Genaueres später).

Die zweite Zeile enthält, da sich kein File auf der Diskette befindet, die Meldung »664 BLOCKS FREE«.

Da sich diese Informationen auf der schon erwähnten Spur 18 befinden, wollen wir uns diese Spur mit EDDI gleich einmal etwas genauer ansehen. Laden Sie den Editor und legen Sie unsere »Spieldiskette« ein; danach starten Sie mit RUN.

Als Kommando tippen Sie <F3> für »BLOCK LESEN«.



Danach geben Sie, durch Komma getrennt, die Spur und Sektornummer des gewünschten Blocks ein; in unserem Fall »18,0«.

Nach dem Ladevorgang meldet sich EDDI mit Byte 0 der ersten von 16 Seiten, zu je 16 Byte. Drücken Sie jetzt <RETURN>, um die erste Seite anzuzeigen, welche wir nun betrachten wollen.

Es sollte vielleicht erwähnt werden, daß die Zählung von Blöcken und Bytes grundsätzlich bei Null beginnt. Den geladenen Block bezeichnet man als BAM (Block Availability Map), auf deutsch etwa »Blockbelegungsplan«. Dieser Plan gibt an, welche Blöcke auf der Diskette frei und welche schon beschrieben sind. Ferner enthält er den Namen der Diskette, die ID, das Formatkennzeichen und den Beginn des Directory.

Die ersten beiden Byte (0,1) dieses Blocks enthalten Spur und Sektor des ersten Directory-Blocks; normalerweise »18,1« (siehe auch Tabelle 2). Byte 2 enthält das Formatkennzeichen (hier 65, beziehungsweise »A«). Zur Erklärung: Commodore stellt verschiedene Laufwerke her, zum Beispiel die 1541, 4040, 8050 und 8250. Diese Laufwerke unterscheiden sich fast alle im Aufzeichnungsformat, das heißt Anzahl und Verteilung der Spuren und Sektoren; so hat die CBM 8050 77 Spuren mit bis zu 29 Sektoren, was deren höhere Speicherkapazität zur Folge hat. Solche Disketten können verständlicherweise von der 1541 weder gelesen noch beschrieben werden. Am Formatkennzeichen »A« erkennt die 1541 nun Disketten ihres eigenen Formats; ist dieses nicht identisch, so beschwert sich die Floppy-Station mit einer Fehlermeldung. Eine Ausnahme dieser Regel bildet die Lesekompatibilität, die besagt, daß eine »fremde« Diskette zwar gelesen, aber nicht beschrieben werden kann (zum Beispiel eine Diskette des 4040- auf dem 1541-Laufwerk).

Byte 3 steht generell auf Null, da es bei der 1541 keine Funktion erfüllt.

Erste Versuche mit EDDI, dem Disk-Monitor/Editor

Die Byte 4 bis 143 enthalten nun die eigentliche BAM, deren Format ein wenig kompliziert ist: Für jede Spur sind 4 Byte reserviert, wobei das jeweils erste Byte die Anzahl der noch freien Blöcke auf dieser Spur angibt. Die folgenden 3 Byte müssen wir als eine Gesamtheit von 24 Bit betrachten, wobei jedes gesetzte Bit einen freien Block signalisiert; siehe auch Tabelle 3.

Um auch die folgenden Seiten des Blocks zu betrachten, drücken Sie zum Vorwärtsblättern <F1>; die weitere Bedienung ist analog zur oben beschriebenen. Rückwärtsblättern ist durch Drücken von <F2> möglich.

Fahren Sie nun bis zum Byte 144 vor und sehen Sie sich die Seite an.

Die Byte 144 bis 161 enthalten den Namen der Diskette, der beim Formatieren festgelegt wird. Direkt im Anschluß daran folgen die Byte 162,163, die die ID im ASCII-Code beinhalten, gefolgt von einem »SHIFT SPACE«. An der ID erkennt die Floppystation, ob die Diskette gewechselt wurde; deshalb sollte jede Diskette eine andere ID haben.

Spur 01 bis 17	21 Sektoren
Spur 18 bis 24	19 Sektoren
Spur 25 bis 30	18 Sektoren
Spur 31 bis 35	17 Sektoren
Spur 36 bis 52	21 Sektoren (nur 1571)
Spur 53 bis 59	19 Sektoren (nur 1571)
Spur 60 bis 65	18 Sektoren (nur 1571)
Spur 66 bis 70	17 Sektoren (nur 1571)

Tabelle 1. Spuren und Sektoren des 1541-Diskettenformates

```
10 REM EDDI - DISKMONITOR/EDITOR                <150>
50 PRINT"{CLR,BLACK}":POKE 53280,14:POKE 5
   3281,14                                      <079>
60 GOSUB 10000                                  <040>
70 OPEN 1,8,15,"I0":OPEN 2,8,2,"#"              <106>
80 PRINT"{CLR,SPACE}E D D I{2SPACE}-{2SPAC
   E}HAUPTMENUE"                                <069>
85 HE$="BYTE{6SPACE}DEC{3SPACE}HEX{3SPACE}
   BIN{8SPACE}ASC":POKE 650,128                 <106>
90 PRINT" EEEEEEEEEEEEEEEEEEEEEE"               <008>
100 PRINT:PRINT:PRINT                           <225>
110 PRINT"(F1) - SCROLLING VORWAERTS":PRIN
    T                                           <147>
120 PRINT"(F2) - SCROLLING RUECKWAERTS":PR
    INT                                         <144>
130 PRINT"(F3) - BLOCK LESEN":PRINT             <078>
140 PRINT"(F4) - BLOCK SCHREIBEN":PRINT         <132>
150 PRINT"(F5) - EDITOR EINSCHALTEN":PRINT      <182>
160 PRINT"(F6) - DISKETTE WECHSELN":PRINT       <094>
170 PRINT"(F7) - RUECKKEHR INS MENUE":PRIN
    T                                           <215>
180 PRINT"(F8) - PROGRAMMENDE"                  <037>
190 PO=1:GOTO 9000                              <246>
1000 REM EDDI AN                                <098>
1010 X=0:Y=0                                    <023>
1020 FOR Y=E TO 255 STEP 16                     <024>
1030 PO=2:PRINT"{CLR}EDITOR-MODUS FUER TRA
     CK"T" SEKTOR"S                             <138>
1040 PRINT:PRINT HE$:PRINT                      <193>
1050 FOR X=Y TO Y+15:PRINT X:NEXT X             <145>
1060 PRINT"{HOME,3DOWN}":FOR X=Y TO Y+15        <043>
1065 DA=PEEK(50000+X):GOSUB 7030:PRINT X,O
     U$                                         <208>
1070 INPUT"{UP,8RIGHT}";IN$:IF IN$=""THEN
     1090                                       <132>
1072 IF LEFT$(IN$,1)="↑"THEN PRINT"{HOME,1
     9DOWN}":GOTO 9000                          <207>
1073 IF LEFT$(IN$,1)="←"THEN PRINT"{HOME,2
     0DOWN}":GOTO 1125                          <196>
1075 DA=VAL(LEFT$(IN$,3)):IF DA>255 OR DA<
     0 THEN PRINT"{2UP}":GOTO 1065              <228>
1080 POKE 50000+X,DA                            <180>
1120 NEXT X:PRINT                               <010>
1125 PRINT"EINGABE ?";                          <243>
1130 GET A$:IF A$=""THEN 1130                   <228>
1140 IF A$="{F1}"THEN 1200                      <101>
1150 IF A$="{F2}"THEN 1300                      <121>
1160 IF A$<>" "THEN NEXT Y                      <066>
1170 PO=1:GOTO 9000                             <210>
1200 PRINT"{HOME,3DOWN}":PRINT E".....???"      <179>
1210 GET A$:IF A$=""THEN 1210                   <115>
1215 IF A$="{F2}"THEN 1300                      <186>
1220 IF A$<>"{F1}"THEN 1020                     <126>
1230 E=E+16:IF E>255 THEN E=0                   <214>
1240 GOTO 1200                                  <196>
1300 PRINT"{HOME,3DOWN}":PRINT E".....???"      <025>
1310 GET A$:IF A$=""THEN 1310                   <025>
1315 IF A$="{F1}"THEN 1200                      <022>
1320 IF A$<>"{F2}"THEN 1020                     <232>
1330 E=E-16:IF E<0 THEN E=240                   <020>
1340 GOTO 1300                                  <058>
2000 REM DISKETTENWECHSEL                       <082>
2010 PRINT"{CLR}BITTE NEUE DISKETTE EINLEG
     EN"                                        <171>
2020 GET A$:IF A$=""THEN 2020                   <196>
2030 RUN                                        <038>
3000 REM BLOCK READ                             <067>
3010 PO=2:PRINT"{CLR,3SPACE}BLOCK LESEN":P
     RINT:PRINT                                 <093>
3020 INPUT"TRACK, SEKTOR ";T,S                  <092>
3025 IF T<1 OR T>35 THEN 3010                   <051>
3030 PRINT#1,"U1 2 0"T;S                        <163>
3035 IF ST<>0 THEN PRINT:GOTO 9000              <195>
3040 PRINT#1,"B-P 2 0"                          <146>
3050 SYS 49152:E=0:X=0:Y=0:GOTO 5010            <188>
3060 FOR Y=E TO 255 STEP 16                     <032>
3070 PRINT"{CLR}TRACK"T" SEKTOR"S               <136>
3080 PRINT:PRINT HE$:PRINT                      <201>
3090 FOR X=Y TO Y+15:DA=PEEK(50000+X):GOSU
     B 7030:PRINT X,OU$:NEXT X                  <138>
3100 GOTO 9000                                  <056>
4000 REM BLOCK WRITE                            <070>
4010 PO=1:PRINT:PRINT:INPUT"{CLR,RED}TRACK
     , SEKTOR";T,S:PRINT"{BLACK}"               <169>
4020 PRINT#1,"B-P 2 0"                          <110>
4030 SYS 49177                                  <024>
4040 PRINT#1,"U2 2 0"T;S                        <030>
4050 GOTO 9000                                  <244>
5000 REM SCROLL FORWARD                         <247>
5010 E=X:IF E>255 THEN X=0:E=0                  <156>
5020 PRINT"{CLR}TRACK"T" SEKTOR"S               <054>
5030 PRINT:PRINT HE$:PRINT                      <117>
5040 DA=PEEK(50000+E):GOSUB 7030:PRINT E,O
     U$                                         <136>
5050 X=X+16                                     <023>
5060 GET A$:IF A$=""THEN 5060                   <030>
5070 IF A$="{F1}"THEN 5010                      <237>
5075 IF A$="{F2}"THEN X=X-16:GOTO 6010          <156>
5077 IF A$="{F5}"THEN 1000                      <213>
5080 GOTO 3060                                  <146>
6000 REM SCROLL BACKWARD                        <055>
6010 E=X:IF E<0 THEN E=240:X=240                <161>
6020 PRINT"{CLR}TRACK"T" SEKTOR"S               <038>
6030 PRINT:PRINT HE$:PRINT                      <101>
6040 DA=PEEK(50000+E):GOSUB 7030:PRINT E,O
     U$                                         <120>
6050 X=X-16                                     <039>
6060 GET A$:IF A$=""THEN 6060                   <046>
6070 IF A$="{F2}"THEN 6010                      <227>
6075 IF A$="{F1}"THEN X=X+16:GOTO 5010          <102>
6077 IF A$="{F5}"THEN 1000                      <197>
6080 GOTO 3060                                  <130>
7000 REM BEREITSTELLUNG DES STRINGS             <077>
7010 REM DA/DA$ SIND AUSGABEWERTE
          H$,D$,B$,C$ SIND ZWISCHENWERTE        <087>
7020 REM OU,OU$ SIND ENDERGEBNISSE              <168>
7030 IF DA>31 AND DA<128 OR DA>159 AND DA<
     256 THEN C$=CHR$(DA):GOTO 7040             <166>
7035 C$="."                                     <234>
7040 XX$="000":D$=RIGHT$(STR$(DA),LEN(STR$
     (DA))-1)                                   <056>
7045 D$=LEFT$(XX$,3-LEN(D$))+D$                 <229>
7050 XX$="123456789ABCDEF":H$=""                <225>
7060 HH=INT(DA/16):HL=DA-HH*16                  <126>
7070 IF HH THEN H$=H$+MID$(XX$,HH,1):GOTO
     7080                                       <163>
7075 H$=H$+"0"                                  <183>
7080 IF HL THEN H$=H$+MID$(XX$,HL,1):GOTO
     7090                                       <021>
7085 H$=H$+"0"                                  <193>
7090 B$="":FOR Q=7 TO 0 STEP-1                  <140>
7100 IF(DA AND(2↑Q))<>0 THEN B$=B$+"1":NEX
     T:GOTO 7110                                <099>
7105 B$=B$+"0":NEXT                             <245>
7110 OU$=D$+"{3SPACE}"+H$+"{4SPACE}"+B$+"{
     3SPACE}"+C$                                <035>
7120 RETURN                                     <064>
8999 END                                        <111>
9000 REM GET KOMMANDO                           <200>
9010 PRINT:PRINT"KOMMANDO ?{2SPACE}";           <171>
9020 PRINT"{LEFT}▒";:FOR W=1 TO 75:GET KO$
     :IF KO$<>""THEN 9090                       <158>
9030 NEXT W                                     <080>
9040 PRINT"{LEFT,RVSON}▒{RVOFF}";:FOR W=1
     TO 75:GET KO$:IF KO$<>""THEN 9090          <220>
9050 NEXT W                                     <100>
9060 GOTO 9020                                  <238>
9090 IF KO$="@"THEN 9200                        <165>
9100 IF ASC(KO$)>140 OR ASC(KO$)<133 THEN
     9020                                       <202>
9110 KO=ASC(KO$)-132                            <210>
9120 ON PO GOTO 9130,9140,20000                 <086>
9130 ON KO GOTO 9020,3000,1000,80,9020,400
     0,2000,20000                               <251>
9140 ON KO GOTO 5000,3000,1000,80,6000,400
     0,2000,20000                               <247>
9200 PRINT                                      <156>
9210 GET#1,A$:PRINT A$;:IF ST<>64 THEN 921
     0                                          <082>
9220 GOTO 9000                                  <080>
9999 END                                        <095>
10000 DATA 160,0,169,8,32,9,237,169,98,32,
      199,237,32,19,238,153,80,195,200          <054>
10010 DATA 208,247,32,239,237,96,160,0,169
      ,8,32,12,237,169,98,32,185,237            <155>
10020 DATA 185,80,195,32,221,237,200,208,2
      47,32,254,237,96,0,0                      <054>
10030 RESTORE:FOR Z=1 TO 51:READ A:POKE 49
      151+Z,A:NEXT                              <168>
10040 REM GET:49152; WRITE:49177                <225>
10050 RETURN                                    <202>
20000 PRINT:PRINT:PRINT"{LIG.BLUE}AUF WIED
      ERSEHEN !!!":PRINT:POKE 53280,14:POK
      E 53281,6                                 <099>
20001 PRINT"UND DANKESCHOEN !"                  <133>
```

Listing 1. EDDI, ein Disk-Monitor/Editor

Byte 165 und 166 enthalten DOS-Version und Formatkennzeichen, hier normalerweise »2A«, wiederum gefolgt von einem »SHIFT SPACE«.

Die Bytes 171 bis 255 haben normalerweise keine Bedeutung und können unterschiedlich gefüllt sein.

Wie sieht das Inhaltsverzeichnis aus?

Auf unserer Entdeckungsreise durch Spur 18 folgen wir jetzt der Angabe in den ersten beiden Byte und laden den ersten Directory-Block (<F3>; 18,1). Das Format des Blocks ist der Tabelle 4 zu entnehmen. Jeder Directory-Block enthält acht File-Einträge und den Zeiger auf den nächsten Directory-Block (Byte 0 und 1); ist die Track-Nummer des nächsten Blocks 0, so war der gelesene Directory-Block der letzte, und das zweite Byte zeigt die Anzahl der hier benutzten Byte. In unserem Fall stehen hier 0 und 255.

Nun zu Tabelle 5, die das Format eines Directory-Eintrags darlegt: Jeder dieser Einträge besteht aus 30 Byte, wobei das erste den Filetyp (siehe Tabelle 6), die beiden nächsten Spuren und Sektoren des ersten Fileblocks und die 16 folgenden Bytes den Filenamen enthalten. Die folgenden 3 Byte werden nur bei relativen Dateien verwendet; sie werden später im einzelnen noch besprochen.

Byte 26 und 27 enthalten Track und Sektor des neuen Files, falls das alte mit »@« überschrieben wurde. Die Byte 28, 29 schließlich geben die Anzahl der belegten Blöcke dieses Files an.

Die einzelnen Datei-Typen

Diese bis jetzt beschriebenen Angaben werden vom Betriebssystem der Floppy, also vom DOS (englisch: Disk Operating System), verwaltet.

Beschäftigen wir uns nun mit den restlichen Blöcken auf der Diskette, die dem Anwender zur freien Verfügung stehen, denn dort werden die einzelnen Files gespeichert, deren Aufbau uns jetzt interessiert.

DEL-Files:

Diese Fileanzeige existiert normalerweise nicht im Directory; wird ein File gelöscht, so wird dieses nicht mehr angezeigt; das Byte des Filetyps steht dann auf 0. Durch Setzen des Filetyps auf 128 (hex. $80) kann eine DEL-Anzeige jedoch erzwungen werden.

SEQ-Files:

Dieser Filetyp dient zur Speicherung von Daten auf Diskette (im Gegensatz zur Programmspeicherung). Der Aufbau dieses Filetyps ist relativ einfach: Die ersten beiden Bytes eines Datenblocks zeigen jeweils auf den nächsten Block im File; so erfolgt eine beliebig lange Blockverkettung auf der Diskette. Da aber auch das schönste File einmal zu Ende geht, muß der letzte Block gekennzeichnet sein. Dies erfolgt, wie schon beim Directory, durch eine 0 als Spurnummer. Die Sektornummer bezeichnet jetzt die Anzahl der belegten Datenbytes dieses Blocks. Diese Art der Verkettung von Blöcken wird bei allen Filetypen vorgenommen! Die restlichen 254 Byte jedes Blocks enthalten die Daten.

USR-Files:

USR-Files stimmen im Aufbau exakt mit den SEQ-Files überein, sie haben jedoch noch Zusatzfunktionen im DOS, auf die später eingegangen werden soll.

PRG-Files:

PRG-Files stellen den häufigsten Filetyp dar. Sie dienen der Speicherung von Programmen auf der Diskette und haben nahezu denselben Aufbau wie SEQ-Files. Der einzige Unterschied besteht in den Byte 2 und 3 des ersten Blocks, welche die Startadresse des Programms im Computer ent-

Byte	Bedeutung
000	enthält 18 ($12); Spurnummer für Directory
001	enthält 1 ($01); Startsektor für Directory
002	enthält 65 ($41); Formatkennzeichen »A«
003	Flag für doppelseitige Disketten (1 = doppelseitige Disk, keine Bedeutung im 1541-Modus)
004	Anzahl der freien Blöcke/Sektoren für Spur 1
005–007	Bitmuster der Blockbelegung für Spur 1: Bit = 1 bedeutet »Sektor/Block frei« Bit = 0 bedeutet »Sektor/Block belegt« Byte 005 enthält die Belegung für Sektor 0–7 Byte 006 enthält die Belegung für Sektor 8–16 Byte 007 enthält die Belegung für Sektor 17–23 (Sektor 21–23 sind natürlich nie vorhanden)
008–011	s.o. 004–007 für Spur 2
...	
140–143	s.o. 004–007 für Spur 35
144–159	Diskettenname, der bei der Formatierung angegeben wird; aufgefüllt mit Charactercodes 160 ($a0)
160–163	zweimal 160 ($a0) »SHIFT SPACE«
164	160 ($a0) »SHIFT SPACE«
165–166	$32 und $41 "2A"; Formatangabe der Diskette
167–170	160 ($a0) »SHIFT SPACE«
171–179	$00 bei 1541-Modus; $a0 bei 1570/71-Modus
180–220	0 ($00); nicht benutzter Bereich
221–255	1541/1570: restlicher Bereich nicht verwendet.
bei 1571:	
221–237	Anzahl der freien Blöcke für Spur 36–52
238	Anzahl der freien Blöcke für Spur 53 (immer 0)
239–244	Anzahl der freien Blöcke für Spur 54–59
245–250	Anzahl der freien Blöcke für Spur 60–65
251–255	Anzahl der freien Blöcke für Spur 66–70

Die 1571 enthält zusätzlich noch ein Verzeichnis in Block 53,0:

Byte	Bedeutung
000	enthält 0 ($00)
001–003	s.o. 005–007 für Spur 36
...	
102–104	s.o. 005–007 für Spur 70
105–255	restlicher Bereich nicht verwendet

Tabelle 2. Aufbau und Inhalt der BAM (Block-Belegungs-Plan) in Spur 18, Sektor 0

Aufbau eines 4-Byte-Eintrages in der BAM (eine Spur)

BYTE(s)	Bedeutung:
000	Anzahl der freien Blöcke dieser Spur
001-003	Belegplan der Spur. Jedes Byte ist zuständig für 8 Sektoren: Byte 1 für 0-7 Bit 7 für Sektor 0 Bit 6 für Sektor 1 und so weiter Byte 2 für 8-15 Byte 3 für 16-23

Tabelle 3. Für jede Spur reserviert die BAM 4 Byte

Aufbau eines Directory-Blocks:

BYTE(s)	Bedeutung
000-001	Spur und Sektor des nächsten Directory-Blocks
002-031	Eintrag Nummer 1
032-033	unbenutzt
034-063	Eintrag Nummer 2
064-066	unbenutzt
067-225	Einträge Nummer 3-7 beziehungsweise unbenutzt
226-255	Eintrag Nummer 8

Tabelle 4. Aufbau des Directory der 1541/70/71

halten. Ist diese Adresse gleich der Adresse des Basic-Anfangs, also 2049 ($0801), so können die Programme mit »LOAD"Name",8« geladen werden; dieser Modus ignoriert die Anfangsadresse auf Diskette und lädt die Programme generell an den Basic-Anfang (sogenanntes relatives Laden). Sollen Programme jedoch an anderen Stellen im Speicher stehen, zum Beispiel Maschinenprogramme, so muß diese angegebene Adresse als Startadresse benutzt werden; man lädt hier absolut mit »LOAD"Name",8,1«.

REL-Files:

Dieser Filetyp ist im Aufbau ungleich komplizierter als die eben besprochenen; es soll daher zuerst kurz auf die Arbeitsweise von REL-Files eingegangen werden. Sequentielle Files haben den Nachteil, daß sie praktisch nur aus einem Datensatz bestehen. Sucht man nun, zum Beispiel in einer Kartei, eine bestimmte Hausnummer oder einen bestimmten Namen, so muß der gesamte Datensatz durchgelesen werden, um die entsprechende Stelle zu finden. In einer relativen Datei geht man deshalb einen anderen Weg, um jede Stelle schnell auffinden zu können.

Es existiert eine beliebige Anzahl (zum Beispiel 100) von Datensätzen, wobei alle Datensätze die gleiche Länge haben müssen (maximal 254 Zeichen).

Das DOS legt jetzt einen sogenannten Side-Sektor an, der aus bis zu sechs Blöcken bestehen kann. Diese Blöcke enthalten nun die Zeiger auf sämtliche Datenblöcke, in denen die Datensätze gespeichert sind (1 Datensatz hat maximal 1 Block Länge). Auch hier sind die Datenblöcke wieder durch Zeiger in den Byte 0 und 1 verkettet. Den Aufbau eines Side-Sektor-Blocks zeigt Tabelle 7. Zum besseren Verständnis hier ein kleines Beispiel:

Wir haben eine relative Datei mit 250 Datensätzen à 127 Zeichen. Diese Datei benötigt also 125 Datenblöcke und zwei Side-Sektor-Blöcke. Im Directory-Eintrag finden wir jetzt die schon erwähnten zusätzlichen Byte-Belegungen: Byte 19 und 20 jedes Eintrags enthalten jetzt Spur und Sektor des ersten Side-Sektor-Blocks; Byte 21 gibt die Datensatzlänge (Recordlänge) an.

Wir wollen jetzt auf den 248. Datensatz zugreifen; das DOS arbeitet nun folgendermaßen: Ein Datensatz enthält 127 Byte, das heißt, es passen zwei Datensätze in einen Block; dadurch errechnet sich der Block, auf den jetzt zugegriffen wird, aus (248-1)/2=123,5. (Minus 1, da immer von 0 an gezählt wird.) Da ein Side-Sektor-Block nur 120 Einträge aufnehmen kann, ist der Zeiger auf den Datenblock im Side-Sektor-Block Nummer 2 zu finden. Dieser wird jetzt anhand des Verzeichnisses in Block 1 gelesen und dann auf Zeiger Nummer 3 (Byte 22,23) zugegriffen. Wir kennen also jetzt Spur und Sektor des Blocks, in dem unser Datensatz steht; die Position des ersten Daten-Byte berechnet sich jetzt aus dem Nachkommaanteil der obigen Division (0.5x254=127). Der Datensatz beginnt also beim 127+2=129. Byte.

Der Aufbau von relativen Dateien ist also, wie schon erwähnt, ziemlich kompliziert; diese Art der Datenspeicherung hat aber einige Vorteile gegenüber der »normalen« mit SEQ-Files.

Bedienungshinweise für EDDI, den Disk-Monitor/Editor

Da unserem U-Boot auf dieser schwierigen Fahrt der Sauerstoff ausgegangen ist, wollen wir uns nun erst einmal erholen. Hier noch ein paar Anregungen zur Arbeit mit EDDI: EDDI kann nicht nur Blöcke lesen und anzeigen; Sie können auch Bytes verändern und diesen Block danach wieder abspeichern.

Dazu laden Sie den zu verändernden Block und fahren auf die Seite, die Sie interessiert; hier tippen Sie als Kommando <F5>, und der Editormodus startet. Sie können jetzt Bytes dezimal abändern, indem Sie den jeweils neuen Wert eingeben und <RETURN> drücken. Wollen Sie aus dem Eingabemodus aussteigen, so tippen Sie entweder <RETURN> und können weiterblättern, ohne den Editor zu verlassen, oder Sie tippen <↑> <RETURN>, um in den Kommandomodus zu kommen. Nach einigem Probieren wird Ihnen EDDI sehr schnell vertraut werden; wir gehen auch in den folgenden Abschnitten noch darauf ein.

Wichtig:

Beim Wechseln einer Diskette muß die Funktionstaste <F6> getippt und nach dem Austausch eine Taste gedrückt werden, sonst reagiert das Laufwerk mit einer Fehlermeldung. Diese können übrigens mit <@> abgerufen werden. Das Zurückschreiben eines Blocks auf Diskette erfolgt mit <F4>, wobei Spur und Sektornummer angegeben werden müssen. Hier noch ein paar Vorschläge zum Ausprobieren: Ändern Sie doch einmal auf Ihrer Versuchsdiskette (!) das Formatkennzeichen (Spur 18, Sektor 0, Byte 2 auf 66 statt jetzt 65) und speichern den Block an die gleiche Stelle auf die Diskette zurück. Versuchen Sie nun einmal, ein kleines Programm auf dieser Diskette zu speichern. (Die genauen Vorgänge in der Floppy-Station werden später erläutert.) Oder ändern Sie einmal die Bytes im Directory, die den File-

Aufbau eines Directory-Eintrags:

BYTE(s)	Bedeutung
000	Filetyp, siehe gesonderte Tabelle
001-002	Spur und Sektor des ersten Datenblocks
003-018	Filename, aufgefüllt mit Charactercode 160
019-020	REL-Files: Spur und Sektor des ersten Side-Sektor-Blocks
021	REL-Files: Datensatzlänge
022-025	unbenutzt
026-027	Spur und Sektor beim Überschreiben mit @ (nur Zwischenspeicher)
028-029	Anzahl der von diesem File belegten Blocks

Tabelle 5. Bedeutung der einzelnen Byte des Directory

Aufbau des Filetyp-Bytes

BIT	Bedeutung, in Klammern jeweiliger Inhalt				
0	(0)	(1)	(0)	(1)	(0)
1	(0)=DEL	(0)=SEQ	(1)=PRG	(1)=USR	(0)=REL
2	(0)	(0)	(0)	(0)	(1)
3	unbenutzt				
4	unbenutzt				
5	unbenutzt				
6	(0)=normal; (1) = File kann durch SCRATCH nicht mehr gelöscht werden				
7	(0)= File noch offen (1)= File ordnungsgemäß geschlossen				

Tabelle 6. Die Bedeutung des ersten Bytes eines Directory-Eintrages

Aufbau eines Side-Sektor-Blocks:

BYTE(s)	Bedeutung
000-001	Spur und Sektor des nächsten Side-Sektor-Blocks
002	Nummer des Side-Sektor-Blocks
003	Datensatzlänge
004-005	Spur und Sektor des Side-Sektor-Blocks 1
006-007	Spur und Sektor des Side-Sektor-Blocks 2
008-015	Spur und Sektor der Side-Sektor-Blöcke 3-6
016-017	Spur und Sektor des ersten Datenblocks, für den der Side-Sektor-Block zuständig ist (Datenblock 0)
	Spur und Sektor des zweiten Datenblocks (Nummer 1)
018-255	Spur und Sektor der Datenblocks Nummer 2 bis Nummer 119

Tabelle 7. Relative Dateien benutzen Side-Sektor-Blöcke, um Datensätze gezielt anzuspringen

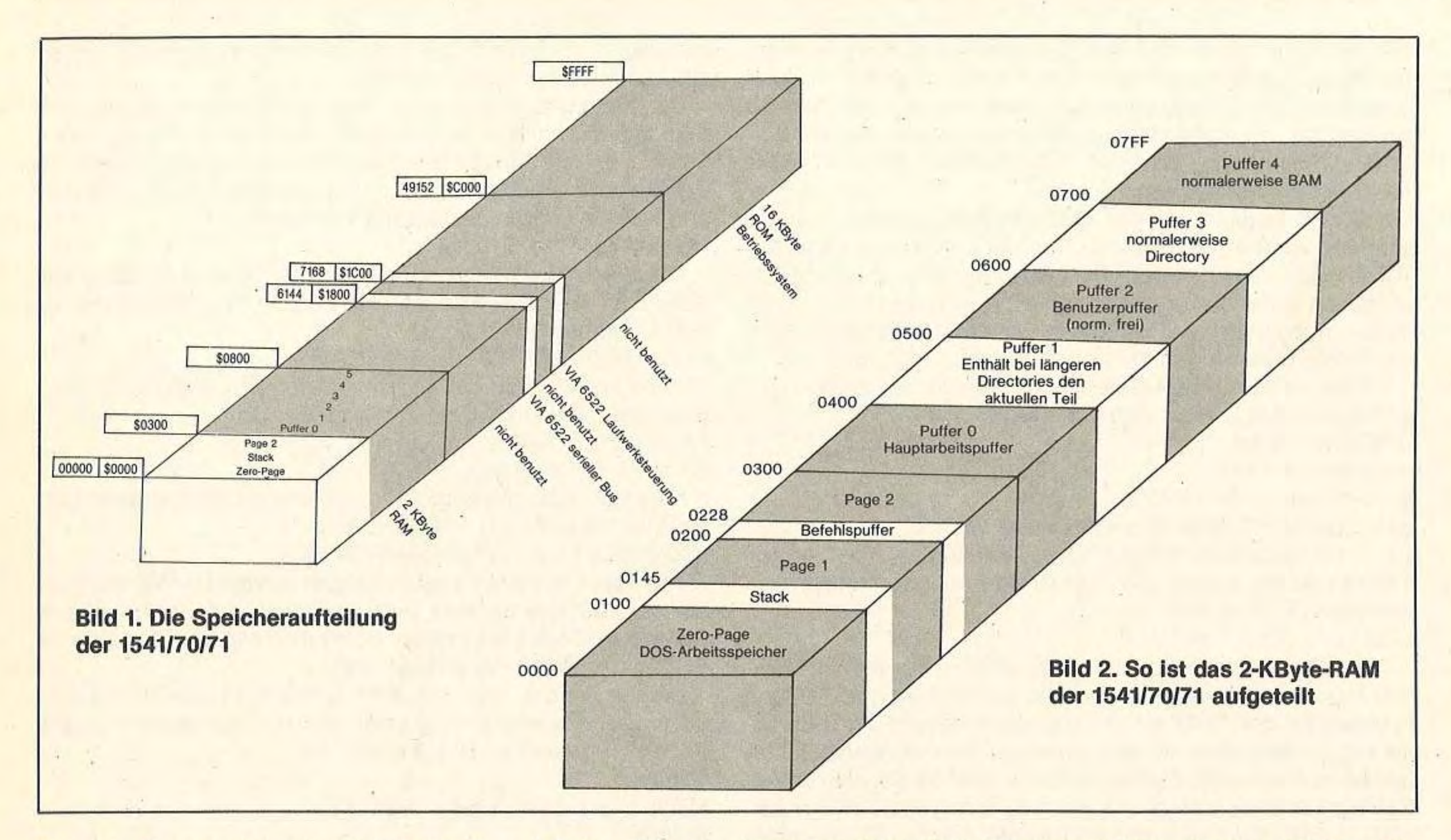


Bild 1. Die Speicheraufteilung der 1541/70/71

Bild 2. So ist das 2-KByte-RAM der 1541/70/71 aufgeteilt

typ angeben, entspechend Tabelle 6 und laden Sie es danach. Experimentieren Sie ruhig ein wenig mit der Floppy-Station. Das wird Ihnen das Verständnis im weiteren Verlauf des Kurses stark erleichtern.

Floppy kontra Datasette

Sicherlich machte sich jeder, der ein schnelleres Peripheriegerät als die Datasette haben wollte, schon seine Gedanken über den Preis der 1541/70/71: »Die kostet ja mehr als der Computer!«. In der Tat ist die 1541-Floppystation von dieser Seite her betrachtet nicht gerade günstig, wer sich jedoch schon intensiver mit ihr beschäftigt hat, wird eine Eigenart festgestellt haben, die sie mit allen anderen Commodore-Laufwerken teilt: Sie ist »intelligent«. Diese Laufwewrke besitzen ein eigenes Betriebssystem (DOS) und eigene Mikroprozessoren. Sie arbeiten völlig unabhängig vom Computer und dessen Speicher. Der Vorteil liegt auf der Hand: Das 1541-Laufwerk beansprucht weder Speicherplatz noch Rechenzeit des Computers, außer beim direkten Datenaustausch. Als Beispiel betrachte man den Befehl »N« (Formatieren). Während der Formatierung steht der Computer zur (fast) freien Verfügung, da dieser Vorgang nur laufwerksintern abläuft und sich der C 64 mit READY meldet, während die 1541 noch arbeitet.

Wir wollen uns jedoch nur den Direktzugriffsbefehlen und den Speicherbefehlen widmen; auch übergehen wir die im Commodore-Handbuch nicht erwähnte relative Datenspeicherung, über die in anderen Ausgaben schon ausführlich gesprochen wurde. Uns sollen nur die Befehle beschäftigen, die uns zur willkürlichen Manipulation von Floppystation und Disketten nützen.

Zur Beruhigung: Ein Beschädigen der 1541 durch direkte Eingriffe in das DOS ist nicht zu befürchten, auch wenn es passieren kann, daß sich das Laufwerk nur noch durch Aus-/Einschalten wieder in den Normalzustand versetzen läßt. Haben Sie übrigens einmal, wie empfohlen, das Formatkennzeichen einer Diskette verändert? Sie werden sicherlich bemerkt haben, daß sich danach nichts mehr auf Ihre Diskette schreiben läßt. Mit diesem Trick, der die gleichen Folgen wie das Anbringen einer Schreibschutzplakette an der Diskette hat, können Sie sich also ganz einfach Ihre Diskette gegen unbeabsichtigtes Löschen sichern. ACHTUNG: Diese Methode funktioniert natürlich nicht, wenn neu formatiert werden soll; dagegen hilft nur das Anbringen einer Schreibschutzplakette!

Die Floppystation verfügt, außer den schon bekannten Befehlen zur Diskettenorganisation, über eine ganze Anzahl weiterer Befehle, mit denen sich ungeahnte Möglichkeiten ergeben, zum Beispiel Herstellen eines eigenen Diskettenformats, Leseschutz von Disketten, Programmschutz, Modifikation der Lade- und Saveroutinen und, und, und. Dafür ist es allerdings nötig, daß wir diese Befehle Schritt für Schritt kennen lernen, bevor wir auf die Tricks der Profis, die Manipulationen des DOS und den gezielten Eingriff in den Programmablauf der Floppystation zu sprechen kommen. Dafür ist allerdings das Beherrschen des C 64 und der Maschinensprache unerläßlich. So lohnt es sich unter Umständen, nachdem man aus dem Basic nichts mehr herausholen kann, den Einstieg in die Assemblerprogrammierung zu wagen. Sehr gute Literatur dafür ist vorhanden. Aber vorerst wollen wir uns noch auf das Basic beschränken, um Sie mit dem Befehlssatz der Floppystation vertraut zu machen.

Befehle an die 1541/70/71

Wie schon erwähnt, handelt es sich bei der 1541 um einen vollständigen Computer, der ebenso wie Ihr C 64 RAM und ein Betriebssystem (DOS) im ROM besitzt.

Die genaue Aufteilung ist in Bild 1 zu sehen. Jetzt soll uns nur der RAM-Bereich interessieren (Bild 2). Nicht nur auf der Diskette, sondern auch im RAM werden Speicherbereiche in Abschnitte zu jeweils 256 Byte aufgeteilt. Sie heißen dann nicht mehr Blocks, sondern Pages (Seiten). Das RAM der 1541/70/71 umfaßt nun genau 8 Pages, durchnumeriert von 0 bis 7, insgesamt also 2 KByte. Die Page Nummer 0 (auch Zero-Page genannt) wird hier, wie auch im C 64, vom Betriebssystem als Arbeitsspeicher benutzt und steht uns des-

halb nicht zur freien Verfügung. Ähnlich verhält es sich mit den Pages 1 und 2. Die Pages 3 bis 7 stellen sogenannte Pufferspeicher dar; hier werden alle Daten, die von der Diskette gelesen beziehungsweise auf sie geschrieben werden, zwischengespeichert, da nur blockweise gelesen oder geschrieben werden kann.

Soll zum Beispiel nur ein einziges Byte auf der Diskette geändert werden, so wird erst der gesamte Block in einen der 5 Pufferspeicher gelesen, dort abgeändert und schließlich komplett wieder zurückgeschrieben. Aus diesen Gründen ist es also notwendig, daß wir uns vor einem Direktzugriff einen der Puffer reservieren, in dem dann gearbeitet wird.

Mit Hilfe des »OPEN«-Befehls eröffnen wir einen Direktzugriffskanal. Die Syntax lautet wie folgt:

OPEN fn, gn, kn, "#"

Hierbei bedeuten:

fn – Filenummer (1-127)
gn – Gerätenummer (normalerweise 8)
kn – Kanalnummer in der Floppy (2-14)

Diese Abkürzungen werden wir im folgenden immer verwenden! Ein Beispiel:

```
OPEN 1, 8, 2, "#"
```

Diese Anweisung öffnet im Computer ein File mit der Nummer 1, adressiert als Gerät die Floppystation (Nummer 8) und reserviert in der 1541 einen Kanal (Nummer 2), dem ein Puffer zugeordnet wird. Mit den laufwerksinternen Kanälen verhält es sich wie folgt: Es stehen insgesamt 16 Kanäle zur Verfügung. Hierbei sind Kanal 0 und 1 für LOAD und SAVE reserviert, Kanal 15 ist der Kommandokanal, den Sie bisher immer benutzt haben, um Befehle (zum Beispiel Formatieren) an das Laufwerk zu senden und die Fehlermeldungen des Laufwerks zu empfangen.

Für unsere Zwecke stehen also noch die Kanäle 2 bis 14 zur Verfügung. In unserem Fall reserviert die Floppystation den nächsten freien Puffer. Will man jedoch einen bestimmten Puffer reservieren, etwa um dort ein Maschinenprogramm abzulegen, so ist es notwendig, der 1541 mitzuteilen, welcher Puffer gewünscht wird:

```
OPEN 1, 8, 2, "#1"
```

Es ist hier allerdings zu beachten, daß der gewählte Puffer nicht schon belegt ist; in diesem Fall gibt die 1541 eine Fehlermeldung aus. Wollen Sie an dieser Stelle mehr über das Auslesen der Fehlermeldungen und deren Bedeutung wissen, können wir Sie hier beruhigt auf das Commodore-Handbuch verweisen.

Im allgemeinen sind Puffer 4 für die BAM und Puffer 3 für das Directory reserviert. Haben Sie die Wahl des Puffers der Floppystation überlassen, so erfahren Sie die gewählte Nummer durch Auslesen des soeben geöffneten Direktzugriffskanals:

```
10 OPEN 1, 8, 2, "#"
20 GET#1, D$
30 D=ASC(D$+CHR$(0))
40 REM PUFFERNUMMER IN D
```

Die BLOCK-Befehle

a) Der BLOCK-READ-Befehl (B-R):

Mit dem BLOCK-READ-Befehl liest man jeden beliebigen Block von Diskette in einen vorher reservierten Puffer. Die Syntax lautet:

PRINT#fn, "B-R"; kn; dn; t; s

dn – Drivenummer (immer 0)
t – Track-Nummer
s – Sektornummer

Beispiel: PRINT#15, "B-R 2 0 18 0"

Diese Befehlsfolge liest den Block 18,0 von der Diskette in den oben reservierten Puffer. Wie man sieht, können anstelle der CHR$-Codes feste Zahlenwerte in den Befehlsstring mit übernommen werden.

Das Ganze hat bloß einen kleinen Schönheitsfehler. Mit dem B-R-Befehl läßt sich das erste Byte eines Blocks nicht lesen. Deshalb benutzt man normalerweise anstatt des B-R-Befehls den U1-Befehl. Dieser hat exakt die gleiche Syntax und kann in jedem Fall benutzt werden:

PRINT#15, "U1 2 0 18 0"

Auf diese User-Befehle kommen wir später zurück. Mit einer GET#-Schleife lassen sich nun die einzelnen Bytes in den Computer einlesen.

b) Der BLOCK-WRITE-Befehl (B-W):

Hiermit lassen sich die Daten aus dem reservierten Puffer wieder auf die Diskette schreiben. Syntax:

PRINT#fn, "B-W"; kn; dn; t; s

Beispiel: PRINT#15, "B-W 2 0 18 0"

Natürlich gibt es analog zum B-W- einen USER-Befehl: U2. Beispiel: PRINT#15, "U2 2 0 18 0".

c) Der BUFFER-POINTER-Befehl (B-P):

Für jeden Puffer gibt es einen Zeiger, den Buffer-Pointer. Dieser zeigt auf das aktuelle Byte im Puffer und wird bei jedem Datenzugriff um Eins erhöht, damit man alle 256 Byte eines Blocks der Reihe nach lesen kann.

Dieser Pointer wird mit dem B-P-Befehl gezielt auf bestimmte Bytes positioniert, wenn man nur einzelne Werte und nicht den gesamten Block lesen will.

Syntax:

PRINT#fn, "B-P"; kn; position

Beispiel:

Wir möchten in die Variable A den Wert des 123. Bytes von Block 1;16 einlesen:

```
10 OPEN 15, 8, 15
20 OPEN 1, 8, 2,"#"
30 PRINT#15, "U1 2 0 1 16"
40 PRINT#15, "B-P 2 122"
50 GET#1, A$
60 A=ASC(A$+CHR$(0))
```

Als weiteres Beispiel dient Listing 1.

d) Der BLOCK-ALLOCATE-Befehl (B-A):

Wenn Sie im Direktzugriffsverfahren eine Diskette beschreiben, muß in der BAM danach auch verzeichnet werden, daß die entsprechenden Blocks mit Daten gefüllt sind und nicht mehr überschrieben werden dürfen. Dazu dient der B-A-

```
100 REM AENDERUNG VON ID, FORMATKENN-
101 REM ZEICHEN & LEERZEICHEN ZWISCHEN
102 REM DIESEN BEIDEN.(INSG. 5 ZEICHEN)
103 REM BSP: ALTE ID :XY 2A
104 REM             ID^  ^FORMATKENNZ.
105 REM KANN AUF      :HALLO
106 REM GEAENDERT WERDEN. DAS LEERZ.
107 REM WIRD HIER ZUM ERSTEN 'L'
108 REM WIRKT SICH NUR AUF DIRECT. AUS!
109 :
110 OPEN 15,8,15,"I":OPEN1,8,2,"#"
120 PRINT#15,"U1 2 0 18 0"
130 PRINT#15,"B-P 2 162"
140 GET#1,A$,B$,C$,D$,E$
150 PRINT A$;B$;C$;D$;E$
160 INPUT"NEU:";N$
170 PRINT#15,"B-P 2 162"
180 PRINT#1,N$;
190 PRINT#15,"U2 2 0 18 0"
200 PRINT#15,"I"
210 CLOSE 8:CLOSE 15
READY.
```

Listing 2. Änderung der ID und des Formatkennzeichens

```
1000 REM UNTERPROGRAMM 1
1001 REM LESEN EINES EINTRAGES AUS DEM
1002 REM DIRECTORY (ALLE 30 BYTES !!!)
1003 REM IN DIE VARIABLE DD$
1004 REM UEBERGABEPARAMETER:
1005 REM MM=NUMMER DES EINTRAGES DER
1006 REM     GELESEN WERDEN SOLL
1007 :
1008 :
1009 :
1010 OPEN 15,8,15,"I":OPEN8,8,8,"#"
1020 NN$="":FORI=1TO30:NN$=NN$+CHR$(0):N
EXTI
1030 XX=INT((MM-1)/8)
1040 PRINT#15,"U1 8 0 18 0"
1050 FORZZ=1TOXX+1
1060 PRINT#15,"B-P 8 0"
1070 GET#8,TT$:TT=ASC(TT$+CHR$(0))
1080 GET#8,SS$:SS=ASC(SS$+CHR$(0))
1090 IF TT=0 THEN DD$=NN$:GOTO1170
1100 PRINT#15,"U1 8 0";TT;SS
1110 NEXTZZ
1120 PP=MM-(XX*8):PP=(PP-1)*32+2
1130 PRINT#15,"B-P 8";PP
1140 FORZZ=1 TO 30:GET#8,ZZ$
1150 IFZZ$=""THENZZ$=CHR$(0)
1160 DD$=DD$+ZZ$:NEXTZZ
1170 CLOSE 8:CLOSE 15
1180 RETURN
READY.
```

Listing 3. Unterprogramm 1. Lesen eines Eintrages aus dem Directory.

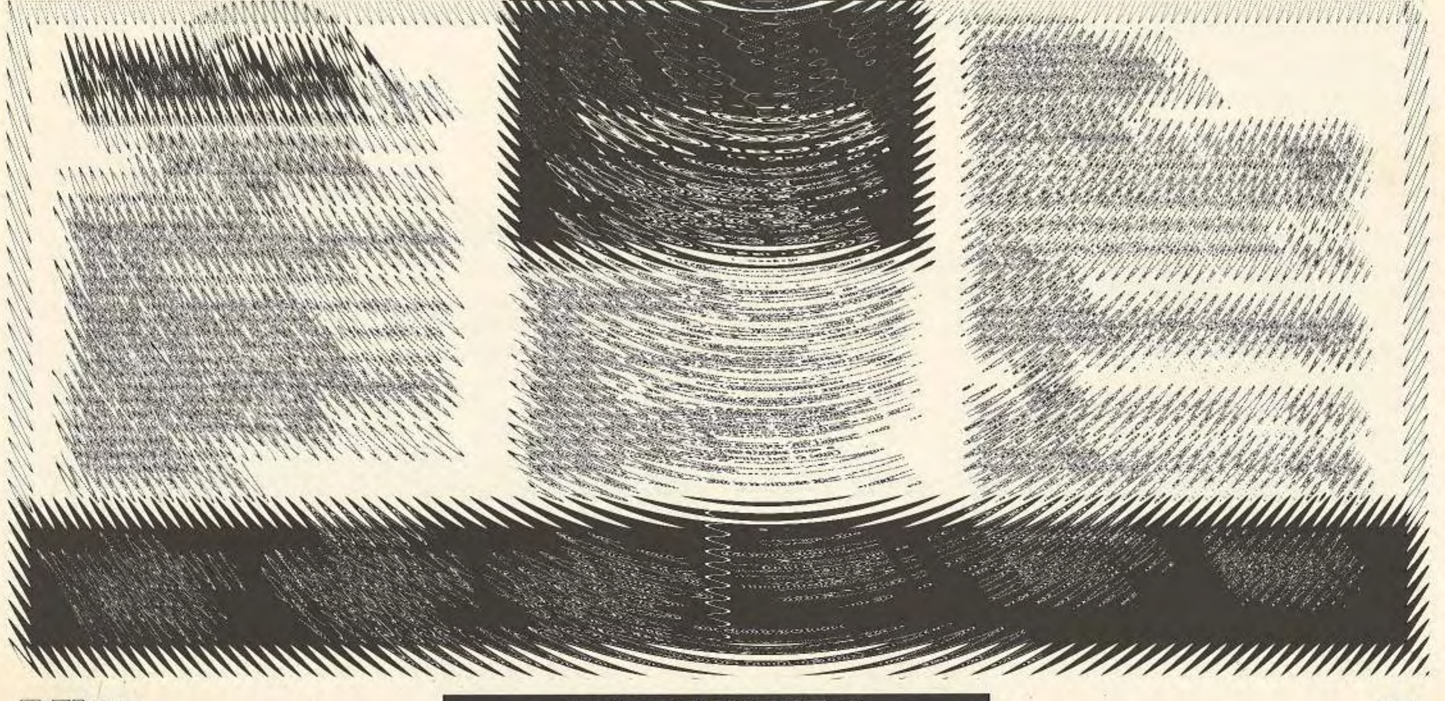
```
2000 REM UNTERPROGRAMM 2
2001 REM SCHREIBEN EINES EINTRAGES IN
2002 REM DAS DIRECTORY (30 BYTES !!!)
2003 REM UEBERGABEPARAMETER:
2004 REM MM=NUMMER DES EINTRAGES DER
2005 REM     GESCHRIEBEN WERDEN SOLL
2006 REM DD$=DIRECTORYEINTRAG
2007 :
2008 :
2009 :
2010 OPEN 15,8,15,"I":OPEN8,8,8,"#"
2020 XX=INT((MM-1)/8)
2030 PRINT#15,"U1 8 0 18 0"
2040 FORZZ=1TOXX+1
2050 PRINT#15,"B-P 8 0"
2060 GET#8,T$:TT=ASC(T$+CHR$(0))
2070 GET#8,S$:SS=ASC(S$+CHR$(0))
2080 IF TT=0 THEN 2150
2090 PRINT#15,"U1 8 0";TT;SS
2100 NEXTZZ
2110 PP=MM-(XX*8):PP=(PP-1)*32+2
2120 PRINT#15,"B-P 8";PP
2130 PRINT#8,DD$;
2140 PRINT#15,"U2 8 0";TT;SS
2150 CLOSE 8:CLOSE 15
2160 RETURN
READY.
```

Listing 4. Unterprogramm 2. Schreiben eines Directory-Eintrages

Befehl, der jeden beliebigen Block in der BAM als belegt kennzeichnet. Die Syntax lautet:
PRINT#fn, "B-A"; dn; t; s
Beispiel:
PRINT#15, "B-A 0 1 16"
kennzeichnet Block 1;16 als belegt; war dieser Block schon belegt, meldet sich die Floppystation mit der Fehlermeldung »65, NO BLOCK, XX, YY«; wobei XX und YY die Track- und Sektornummer des nächsten freien Blocks angeben.
e) Der BLOCK-FREE-Befehl (B-F):
Dieser ist das genaue Gegenstück zum B-A-Befehl; er deklariert einmal belegte Blöcke wieder als frei für einen weiteren Zugriff. Seine Syntax ist identisch mit der des B-A-Befehls.
f) Der BLOCK-EXECUTE-Befehl (B-E):
Dieser Befehl nimmt eine Sonderstellung ein. Er gleicht im Prinzip dem B-R-Befehl; nur mit dem zusätzlichen Effekt, daß der eingelesene Block im Puffer als Maschinenprogramm gestartet wird.
Zur Vertiefung der Block-Befehle sei noch auf die Listings 2 bis 7 hingewiesen, welche die eben besprochenen Anwendungen noch an praktischen Beispielen verdeutlichen.

Die MEMORY-BEFEHLE

a) Der MEMORY-READ-Befehl (M-R):
Dieser Befehl entspricht haargenau dem PEEK-Befehl in Basic. Mit ihm können Sie jede beliebige Speicherstelle der Floppystation auslesen.



```
100 REM BEISPIEL FUER EINE KLEINE
101 REM DIRECTORY-MANIPULATION:
102 REM SCRATCH-SCHUTZ EINZELNER FILES
103 REM NACH ANZEIGE DES FILENAMENS:
104 REM J = SCHUETZE DIES FILE
105 REM N = WEITER ZUM NAECHSTEN FILE
106 REM E = ENDE
107 REM ACHTUNG !!! "SCHUETZT" AUCH
108 REM SCHON GESCRATCHTE FILES WENN
109 REM VERLANGT, STELLT SIE ABER NICHT
110 REM WIEDER HER !!!
111 REM SCRATCH-SCHUTZ WIRD IM DIRECT.
112 REM DURCH EIN '<' HINTER DEM
113 REM FILETYP ANGEZEIGT. NAEHERES
114 REM SIEHE TABELLE FOLGE 1 !!!
115 REM ACHTUNG !!! NUR ZUSAMMEN MIT
116 REM DEN UNTERPROGRAMMEN 1 & 2
117 REM LAUFFAEHIG !!!
118 :
119 :
120 MM=0
130 MM=MM+1:DD$="":GOSUB1000
140 IF DD$=NN$THENEND
150 PRINTMID$(DD$,4,16):INPUTAA$
160 IF AA$="E"THEN END
170 IF AA$="N"THEN 130
180 HH$=LEFT$(DD$,1)
190 HH$=CHR$(ASC(HH$)OR2^6)
200 DD$=HH$+RIGHT$(DD$,29)
210 GOSUB2000
220 GOTO 130
230 END
READY.
```

Listing 5. So kann man Files schützen

Syntax:
PRINT#fn, "M-R"; CHR$(adl); CHR$(adh); CHR$(n)
adl = Low-Byte
adh = High-Byte
n = Anzahl (0 bis 255)

Abgeholt werden die gelesenen Daten ebenfalls über den Kommandokanal mit GET#.

Beispiel: Lesen der beiden ID-Zeichen im ASCII-Format der zuletzt initialisierten Diskette:

```
10 OPEN 15, 8, 15
20 PRINT#15, "M-R" CHR$(18) CHR$(0) CHR$(2)
30 GET#15, A$, B$
40 PRINT A$; B$
```

Diese Routine liest die Zero-Page-Adressen 18 und 19, in denen die entsprechenden Werte gespeichert sind.

b) Der MEMORY-WRITE-Befehl (M-W):
Dieses Kommando kann als POKE-Befehl in den Floppyspeicher angesehen werden.
Die Syntax ist hier wie folgt:
PRINT#fn, "M-W"; CHR$(adl) CHR$(adh) CHR$(n) CHR$(data1) CHR$(data2)...

c) Der MEMORY-EXECUTE-Befehl (M-E):
Auch dieser Befehl ist äquivalent zu einem Basic-Befehl, dem SYS-Befehl. Mit ihm kann man also ein Maschinenprogramm an einer beliebigen Stelle im Floppyspeicher ausführen.
Syntax:
PRINT#fn, "M-E" CHR$(adl) CHR$ (adh).
Siehe auch Listing 8.

```
100 REM SCHREIBSCHUTZ SETZEN / LOESCHEN
101 REM DURCH AENDERUNG DES FORMAT-
102 REM KENNZEICHENS IN DER BAM !!!
103 REM FUNKTIONSWEISE SETZEN :
104 REM FORMATKENNZEICHEN WIRD AUF
105 REM BELIEBIGEN WERT AUSSER 'A'
106 REM GESETZT. AB SOFORT KOENNEN
107 REM KEINE SCHREIBVORGAENGE AUSSER
108 REM FORMATIEREN DURCHGEFUEHRT
109 REM WERDEN. ALSO VORSICHT !
110 REM FUNKTIONSWEISE FREIGEBEN :
111 REM DIE FLOPPY SPEICHERT DAS
112 REM FORMATKENNZEICHEN DER EINGEL.
113 REM DISKETTE IN DER SPEICHERSTELLE
114 REM $0101 ZWISCHEN. WIRD DIES
115 REM VOR DEM SCHREIBVORGANG AUF
116 REM 'A' ZURUECKGESETZT, LAESST
117 REM SICH DIE FLOPPY 'UEBERLISTEN'
118 REM DAS 'A' WIRD NUN IN DIE BAM
119 REM GESCHRIEBEN
120 :
130 PRINT"WOLLEN SIE DIE EINGEL. DISKETT
E"
140 INPUT"SCHUETZEN ODER FREIGEBEN";AA$
150 IF AA$="S"THEN 200
160 IF AA$="F"THEN 300
170 PRINT:RUN
180 :
190 :
200 OPEN 15,8,15,"I":OPEN8,8,8,"#"
210 PRINT#15,"U1 8 0 18 0"
220 PRINT#15,"B-P 8 2"
230 PRINT#8,"X";
240 PRINT#15,"U2 8 0 18 0"
250 PRINT#15,"I"
260 CLOSE8:CLOSE15:PRINT:RUN
270 :
280 :
290 :
300 OPEN 15,8,15,"I":OPEN8,8,8,"#"
310 PRINT#15,"U1 8 0 18 0"
320 PRINT#15,"B-P 8 2"
330 PRINT#15,"M-W"CHR$(1)CHR$(1)CHR$(1)C
HR$(65)
340 PRINT#8,"A";
350 PRINT#15,"U2 8 0 18 0"
360 PRINT#15,"I"
370 CLOSE8:CLOSE15:PRINT:RUN

READY.
```

Listing 6. Schützen Sie Ihre Diskette vor jedem Schreibzugriff

Die User-Befehle

Die User-Befehle stellen eine Erweiterung des Befehlssatzes dar, der fast ausschließlich der Bequemlichkeit dient. U1 und U2 wurden schon besprochen, sie ersetzen B-R und B-W.

Die Befehle U3 bis U8 dienen zum Starten eines Maschinenprogramms im Floppyspeicher, dessen Anfangsadressen in einer Tabelle abgelegt sind, so entsprechen:
U3 einem Start bei $0500
U4 einem Start bei $0503
...
U8 einem Start bei $050F.
U4 ersetzt also beispielsweise den Befehlsstring: M-E CHR$(3)CHR$(5).

```
100 REM DIRECTORY-SORTER                              <180>
101 REM SORTIERT DIRECTORY ALPHABETISCH               <141>
102 REM BEI VIELEN EINTRAEGEN BITTE                   <226>
103 REM ETWAS GEDULD (MAX. 5.MIN)                     <219>
104 REM SORTIERT AUCH GESCRATCHTE FILES               <052>
105 REM MIT, STELLT SIE ABER NICHT                    <087>
106 REM WIEDER HER ! SORTIERALGORITHMUS               <048>
107 REM KANN SICH IN EINEM SOLCHEN FALL               <121>
108 REM IN EINER ENDLOSSCHLEIFE VER-                  <039>
109 REM HEDDERN. ABHILFE: NACH 3-4 MIN.               <009>
110 REM STOP-TASTE DRUECKEN, DANN                     <143>
111 REM GOTO 210 EINGEBEN. SIND EINTR.                <019>
112 REM DANN NOCH NICHT VOLLKOMMEN SOR-               <227>
113 REM TIERT, NOCHMALS FUER EINIGE                   <236>
114 REM MINUTEN LAUFEN LASSEN.                        <208>
115 REM ACHTUNG !!! NUR ZUSAMMEN MIT                  <190>
116 REM DEN UNTERPROGRAMMEN 1 & 2                     <233>
117 REM ABLAUFFAEHIG !!!                              <182>
118 :                                                 <176>
119 :                                                 <177>
120 DIM DD$(144)                                      <148>
130 MM=MM+1:GOSUB 1000                                <203>
140 IF DD$=NN$THEN MM=MM-1:GOTO 160                   <248>
150 DD$(MM)=DD$:DD$="":GOTO 130                       <190>
160 FOR GG=1 TO MM-1                                  <172>
170 IF MID$(DD$(GG),4,16)<MID$(DD$(GG+1),4,
    16)THEN 190                                       <054>
180 HH$=DD$(GG):DD$(GG)=DD$(GG+1):DD$(GG+1)=HH$
    :FF=1                                             <049>
190 NEXT GG                                           <206>
200 IF FF THEN FF=0:GOTO 160                          <078>
210 II=MM                                             <176>
220 FOR MM=1 TO II:DD$=DD$(MM):GOSUB 2000
    :NEXT MM                                          <030>
230 END                                               <102>
```

Listing 7. Eine einfache Routine, um das Directory zu sortieren

```
100 REM BEISPIEL FUER MEMORY-EXECUTE
101 REM LOEST IN DER FLOPPY LANGSAMES
102 REM BLINKEN DER ROTEN LED (KENN-
103 REM ZEICHNET NORMALERWEISE HARD-
104 REM WARE-FEHLER) AUS.
105 REM KANN NUR DURCH AUSLOESEN EINES
106 REM RESETS ENTWEDER DURCH EIN/AUS-
107 REM SCHALTEN DER FLOPPY ODER DES
108 REM COMPUTERS BEENDET WERDEN.
109 REM EINSPRUNGSADRESSE : $EA6E
110 :
120 OPEN15,8,15
130 PRINT#15,"M-E"CHR$(110)CHR$(234)
140 CLOSE 15
150 END
READY.
```

Listing 8. Simulieren Sie einen Hardware-Fehler

Auflistung aller verwendeten ROM-Routinen

Kürzel	Adresse		
		SECTLK	$FF96
		SECLST	$FF93
FILPAR	$FFBA	IECOUT	$FFA8
FILNAM	$FFBD	IECIN	$FFA5
OPEN	$FFC0	FILTAB	$F30F
CLOSE	$FFC3	FILSET	$F31F
LISTEN	$FFB1	LOAD	$FFD5
UNLIST	$FFAE	SAVE	$FFD8
TALK	$FFB4	BASOUT	$FFD2
UNTALK	$FFAB	CLALL	$FFE7

Tabelle 8. Die im Artikel erwähnten Betriebssystemroutinen

U9 zeigt auf den NMI-Vektor der 1541, welcher allerdings eine Sonderfunktion hat: Mit U9+ wird die Floppystation auf C64- und mit U9- auf VC20-Betrieb umgeschaltet.
U: stellt einen Reset dar, ähnlich dem SYS 64738 beim C64.

Mit den Kenntnissen über den Befehlssatz der 1541 dürfte es Ihnen nun keine Schwierigkeiten mehr bereiten, sich das Programm EDDI einmal zu Gemüte zu führen. Das einzig Besondere daran sind die Routinen zum Lesen und Schreiben eines Blocks, die aus Geschwindigkeitsgründen in Maschinensprache geschrieben sind. Ein großer Teil der in diesen Folgen erwähnten Informationen ist auch im Commodore-Handbuch enthalten, nur sind dort oft Fehler vorhanden.

1541/70/71 und Assembler

Wenn wir im folgenden von Routinen sprechen, die im Betriebssystem stehen, so werden wir die in Tabelle 8 dargestellten Kürzel verwenden, die Sie übrigens auch in Editorprogrammen gut benutzen können.

FILPAR und FILNAM

Bei OPEN, LOAD und ähnlichen Befehlen müssen Sie entsprechenden Routinen mitteilen, welches File Sie wo öffnen wollen. Um Ihnen eine »Herumwurstelei« in der Zero-Page zu ersparen, wo Sie die einzelnen Angaben von Hand setzen müßten, hat das Betriebssystem zwei entsprechende Routinen implementiert. FILPAR setzt für Sie die einzelnen Fileparameter. Diese müssen der Routine in den Prozessorregistern übergeben werden:

- Filenummer (Akku)
- Geräteadresse (X-Register)
- Sekundäradresse (Y-Register)

Ein Beispiel:
Sie wollen für ein File mit der Nummer 1, der Geräteadresse 8 und der Sekundäradresse 15 (Kommandokanal) die entsprechenden Fileparameter setzen:

```
LDA   #$01     ; Filenummer 1
LDX   #$08     ; Geräteadresse 8
LDY   #$6F     ; Sekundäradresse + $60
JSR   #FILPAR  ; Fileparameter setzen
```

Wie Sie sehen, muß zu der betreffenden Sekundäradresse ein Wert von $60 addiert werden.

Aber in vielen Fällen müssen Sie ja auch einen Filenamen angeben. Dazu dient die FILNAM-Routine. Hier erfolgt die Parameterübergabe:

- Länge des Filenamens (Akku)
- Adresse LO des Namens (X-Register)
- Adresse HI des Namens (Y-Register)

Und wieder ein Beispiel. Um das Directory-File mit dem Namen »$« anzusprechen, geben Sie bitte folgende Befehle ein:

```
LDA   #$24     ; Code für '$' in Akku
STA    $FF     ; und speichern
LDA   #$01     ; Länge des Filenamens
LDX   #$FF     ; Adresse LO
LDY   #$00     ; Adresse HI
JSR   FILNAM   ; übergeben
```

Sie müssen also wissen, wo der Filename im Speicher steht und wie lang er ist. Dies ist aber im allgemeinen kein Problem. Auf die gleiche Weise können Sie der Floppystation über den Kommandokanal auch Befehle senden, wie Sie in der letzten Folge vorgestellt wurden. Das entspräche der Basic-Sequenz:
OPEN x, 8, 15, "befehl"

Natürlich können Sie auch alle Parameter von Hand setzen, beziehungsweise noch einmal lesen. Wo sich die einzelnen

Parameter in der Zero-Page nach Ausführung dieser und der anderen Routinen befinden, ist in Tabelle 9 angegeben.

OPEN und CLOSE

Nachdem wir alle Fileparameter und den Filenamen übergeben haben, können wir die OPEN-Routine mit JSR OPEN aufrufen. Schon ist das entsprechende File geöffnet. Zu beachten wäre folgendes: Es können im Computer niemals mehr als 10 Files gleichzeitig geöffnet sein!

Die CLOSE-Routine arbeitet analog zu OPEN, mit der Ausnahme, daß nur die Filenummer übergeben werden muß. Geräteadresse und Sekundäradresse sucht sich der C 64 aus einer Tabelle heraus, auf die wir später noch zu sprechen kommen:

```
LDA   #$01     ; Filenummer
JSR   CLOSE      in Akku
```

Der Filename wird beim Schließen überhaupt nicht mehr benötigt.

LISTEN und UNLISTEN, TALK und UNTALK

Nach dem Öffnen eines Files kann die Datenübertragung noch nicht beginnen. Sie müssen dem entsprechenden Gerät zuerst mitteilen, ob es senden oder empfangen soll.

Bestes Beispiel ist wieder der Kommandokanal. Über diesen kann das Floppy-Laufwerk sowohl Befehle empfangen, als auch Fehlermeldungen senden.

Um ein Gerät zum Empfangen zu veranlassen, verwenden wir die Routine LISTEN. Das hat nichts mit dem Basic-Befehl LIST zu tun, sondern kommt vom englischen Wort für »Hören«. Beim Aufruf von LISTEN ist das angesprochene Gerät auf Empfang und der Computer auf Senden eingestellt.

Wichtig ist, daß der Akku beim Aufruf die Geräteadresse enthält. Dies gilt für alle vier hier beschriebenen Routinen. Wenn Sie mit dem Senden der Daten fertig sind, sollten Sie ein UNLISTEN zum entsprechenden Gerät schicken, um dieses wieder freizugeben. Dies geschieht mit Hilfe der UNLIST-Routine. Analog verhält es sich mit den Routinen TALK und UNTALK. Sie veranlassen das angesprochene Gerät, Daten zu senden, beziehungsweise mit dem Senden aufzuhören und wieder in den Wartezustand zurückzukehren.

SECTLK und SECLST

Die beiden Routinen SECTLK und SECLST sind ebenfalls sehr wichtig für die Datenübertragung. Denn obwohl wir beim OPEN-Befehl eine Sekundäradresse angeben, muß diese bei jeder weiteren Übertragung nochmals an das aktuelle Gerät gesendet werden.

Dies hat zwei Gründe: Einerseits können Sie ja mehrere Floppy-Kanäle gleichzeitig geöffnet halten. Damit die Floppystation nun weiß, für welchen Kanal der nächste Schwung von Daten bestimmt ist, beziehungsweise welcher Kanal senden soll, muß nach dem Aufruf von TALK SECTLK, beziehungsweise nach dem Aufruf von LISTEN SECLST durchgeführt werden. Außerdem merkt sich der Computer zwar die angegebene Sekundäradresse, sendet sie aber nicht.

Dies hat praktische Gründe, wie wir noch bei den LOAD/SAVE-Routinen sehen werden. SECTLK und SECLST benötigen die jeweilige Sekundäradresse +$60 im Akku. Diese kann, wie in unseren Beispielen, direkt geladen oder aber auch der entsprechenden Zero-Page-Adresse entnommen werden.

IECOUT und IECIN

Nachdem wir nun endlich alle Vorbereitungen getroffen haben, können wir munter Bytes von der Floppystation zum Computer und umgekehrt übertragen. Dies ist mit den ROM-Routinen denkbar einfach. IECOUT überträgt das im Akku befindliche Byte an das aktuelle Gerät; IECIN empfängt eines und legt es im Akku ab.

Bei aller Sorgfalt - Fehler können immer auftreten, so auch beim Busbetrieb. Um einen in einer Busroutine aufgetretenen Fehler zu signalisieren, verwendet das Betriebssystem das Carry-Flag. Generell gilt: Ist das Carry-Flag gesetzt, so ist etwas nicht in Ordnung, und wir sollten das Status-Byte überprüfen. Dieses Status-Byte steht in der Speicherstelle $90. Immer wenn es ungleich Null ist, liegt irgendein Sonderfall vor. Jedes Bit des Status-Bytes hat eine andere Funktion; Tabelle 10 zeigt diese Belegung. Ist zum Beispiel das Bit 7 gesetzt, so ist das angesprochene Gerät entweder nicht vorhanden oder abgeschaltet. In Basic bekämen wir in einem solchen Fall die Meldung »DEVICE NOT PRESENT ERROR«. Interessant ist für uns noch das Bit 6. Ist es gesetzt, so bedeutet das, daß das letzte Byte der angeforderten Informationen übertragen wurde. Dies können wir uns auch in Basic zunutze machen, um beispielsweise die Fehlermeldung der Floppystation auszulesen:

```
10 OPEN 1, 8, 15
20 GET#1, A$: PRINTA$;: IF ST<>64 THEN 20
30 CLOSE 1
```

Wie Sie an diesem Beispiel sehen, ist der Inhalt der Speicherstelle $90 in der Variablen ST enthalten. Vor jeder neuen

Wichtige Zero-Page-Adressen

Adresse	Bedeutung
$90	Status-Flag
$93	Flag für LOAD/VERIFY
$98	Anzahl der offenen Files
$99	Eingabegerät für BASIN
$9A	Ausgabegerät für BASOUT
$B7	Länge Filename
$B8	aktive Filenummer
$B9	Sekundäradresse
$BA	Geräteadresse
$BB/BC	Zeiger auf Filenamen

Tabelle 9. Dies sind Zero-Page-Adressen, unter denen die aktuellen Fileparameter gespeichert werden.

```
(In $FA/$FB muß die Adresse, in $FC die Länge des
Befehlsstrings stehen.)

   LDA #$01       ; Filenummer
   LDX #$08       ; Gerätenummer
   LDY #$6F       ; Sekundäradresse
   JSR FILPAR     ; setzen.
   LDA #$00       ; Länge Filename=0
   JSR FILNAM     ; da kein Filename.
   JSR OPEN       ; File öffnen
   LDA #$08       ; Geräteadresse
   JSR LISTEN     ; auf Empfangen
   LDA #$6F       ; Sekundäradresse
   JSR SECLST     ; senden.
   LDY #$00       ; Zähler auf Null
X  LDA ($FA),Y    ; Befehlsbyte laden
   JSR IECOUT     ; und übertragen
   INY            ; Zähler erhöhen
   CPY #$FC       ; Befehlslänge
   BNE X          ; noch ein Byte?
   LDA #$08       ; Geräteadresse
   JSR UNLIST     ; Sendung beenden
   LDA #$01       ; Filenummer
   JSR CLOSE      ; Schliessen
   RTS
```

Listing 9. So können Befehlsstrings an die Floppy gesendet werden

Prinzip des Lesens des Fehlerkanals mit Ausgabe auf dem Bildschirm

```
  LDA #$00    ; Zurücksetzen des
  STA  $90    ; Status-Flags
  LDA #$01    ; Filenummer
  LDX #$08    ; Geräteadresse
  LDY #$6F    ; Sekundäradresse
  JSR FILPAR  ; setzen.
  LDA #$00    ; Länge Filename=0
  JSR FILNAM  ; setzen.
  JSR OPEN    ; File öffnen
  LDA #$08    ; Geräteadresse auf
  JSR TALK    ; Senden schalten
  LDA #$6F    ; Sekundäradresse
  JSR SECTLK  ; übertragen
X JSR IECIN   ; Byte empfangen
  JSR BASOUT  ; und ausgeben
  BIT  $90    ; Bit 6 Status=0?
  BVC X       ; dann noch ein Byte
  LDA #$08    ; Geräteadresse
  JSR UNTALK  ; Sendung beenden
  LDA #$01    ; Filenummer
  JSR CLOSE   ; und schliessen
  RTS
```

Listing 10. So läßt sich der Fehlerkanal auslesen und anzeigen

Das Status-Flag

Bit	Bedeutung wenn gesetzt
1	Fehler (Zeitüberschreitung) bei IEC-Eingabe
2	Fehler (Zeitüberschreitung) bei IEC-Ausgabe
3-5	nur für Kassettenbetrieb
6	Übertragung ist beendet
7	Gerät meldet sich nicht

Tabelle 10. Für uns wichtige Bits im Status-Flag

Datenübertragung sollten Sie darauf achten, daß das Status-Byte gelöscht wird, da sonst irrtümlich Fehler festgestellt werden könnten. Zur Verdeutlichung des bisher Gesagten dienen die Listings 9 und 10, die jedoch nur Anhaltspunkte geben sollen. Sie sind weder perfekt noch eintippfertig und sollten auf den jeweiligen Bedarf abgestimmt werden.

Bearbeiten mehrerer Files

Sie werden festgestellt haben, daß wir bisher immer nur mit einem einzigen File gearbeitet haben. Was aber, wenn Sie gleichzeitig zwei Files offenhalten müssen, zum Beispiel, um einen Block von Diskette zu lesen. Sie erinnern sich ja, daß wir dazu sowohl den Kommandokanal als auch einen Übertragungskanal benötigen. Wir könnten zwar jeweils, wenn wir den Kanal wechseln wollen, mit CLOSE den alten schließen und mit OPEN den neuen öffnen, aber es geht auch einfacher.

Voraussetzung ist, daß alle benötigten Files schon geöffnet sind. Dann kann mit Hilfe einer, schon erwähnten, Filetabelle zwischen – bis zu 10 – Files beliebig umgeschaltet werden. Diesen Zweck erfüllen die Routinen FILTAB und FILSET.

FILTAB benötigt im Akku die Nummer des Files, auf das Sie umschalten wollen. Die Routine sucht dann in der Filetabelle nach den entsprechenden anderen Parametern. Tritt hier ein Fehler auf, weil das File noch gar nicht geöffnet wurde, so wird das Zero-Flag gelöscht und es kann mit BNE auf einen Fehler überprüft werden.

FILSET schreibt dann die gefundenen Parameter in die entsprechenden Zero-Page-Adressen. Die komplette Routine zum Umschalten auf das File x lautet also:

```
LDA   #$ xx    ; Nummer des Files
JSR   FILTAB   ; Durchsuchen der Tabelle
BNE   ERROR    ; Fehler?
JSR   FILSET   ; Parameter setzen
```

Die ERROR-Routine müssen Sie natürlich noch selbst schreiben. Danach ist das angewählte File zum aktuellen File geworden. Alle LISTEN, TALK und so weiter beziehen sich jetzt auf dieses neue File.

In den Zero-Page-Adressen aus Tabelle 9 stehen nun die für dieses File aktuellen Parameter, da sie aus der großen Filetabelle automatisch übertragen werden. Eine Ausnahme bildet hier der Filename, da er nur beim Öffnen des Files benötigt wird.

Diese große Filetabelle befindet sich übrigens an den Speicherstellen $0259 bis $0276.

Denken Sie immer daran, vor einem erneuten Umschalten UNLIST oder UNTALK aufzurufen. CLOSE braucht dagegen erst aufgerufen zu werden, wenn die Bearbeitung eines Files völlig abgeschlossen ist.

LOAD und SAVE

Prinzipiell könnten Sie mit dem bisher Erwähnten auch schon Programme laden und speichern, allerdings nur sehr mühselig. Da unser Computer das aber schon von selbst beherrscht, geben wir ihm gern diese Arbeit ab.

Betrachten wir zunächst die LOAD-Routine. Auch hier muß wieder eine Vielzahl an Parametern übergeben werden. Mit FILPAR werden Gerätenummer und Sekundäradresse gesetzt. Eine Filenummer braucht nicht gesetzt zu werden. Für die Sekundäradresse gilt folgendes:

Ist sie gleich Null, so wird das Programm an eine, von Ihnen festgelegte, Speicherstelle geladen. Ist sie gleich Eins, so wird das Programm an die Speicherstelle geladen, an der es bei SAVE stand. Der erste Modus wird vom Betriebssystem ausgenutzt, um Programme ab $0800 zu laden, wenn beim LOAD-Befehl keine Sekundäradresse angegeben wird. Prinzipiell kann aber an jede beliebige Adresse geladen werden! Der Filename wird, wie gewohnt, mit FILNAM gesetzt. Vor dem Aufruf der LOAD-Routine treten zwei, uns neue, Parameter hinzu, die wie folgt übergeben werden:

LOAD/VERIFY-Flag (Akku)
Ladeadresse LO (X-Register)
Ladeadresse HI (Y-Register)

Steht beim Aufruf der Routine im Akku 0, so wird geladen. Steht dort hingegen eine 1, so wird ein VERIFY durchgeführt.

Die Startadresse in den X/Y-Registern wird nur beachtet, wenn die Sekundäradresse gleich Null ist. Alles übrige erledigt die LOAD-Routine, und Sie brauchen nur noch deren Ende abzuwarten. Zur Sekundäradresse wäre noch folgendes zu bemerken:

Egal, was Sie für eine Adresse angeben, zur Floppystation wird immer nur 0 gesendet. Wie Sie schon wissen, ist diese Sekundäradresse laufwerksintern für den LOAD-Befehl reserviert und darf nicht ohne weiteres bei OPEN-Befehlen verwendet werden. Nach Beendigung der LOAD-Routine

wird im X und Y-Register die Endadresse des Programms übergeben.

Die SAVE-Routine hat eine etwas kompliziertere Parameterübergabe. FILPAR braucht nur mit der Gerätenummer im X-Register aufgerufen zu werden, da weder Sekundäradresse noch Filenummer benötigt werden. Das Setzen des Filenamens erfolgt normal über FILNAM.

Übergeben werden müssen nun noch Anfangsadresse und Endadresse+1 des zu speichernden Bereichs. Die Anfangsadressen müssen Sie irgendwo in der Zero-Page in der Reihenfolge LO/HI ablegen. Empfehlenswert wären die Adressen $FB/FC, da diese nicht vom Betriebssystem oder Basic benutzt werden. Im Akku muß dann die Adresse des LO-Byte übergeben werden; wenn Sie die Adresse also unter $FB/FC spreichern, muß im Akku $FB stehen.

Die Endadresse übergeben Sie wie folgt:
LO-Byte im X- und HI-Byte im Y-Register. Es muß immer 1 zur Endadresse addiert werden, da sonst das letzte Byte des Programms nicht gespeichert wird. Danach kann die Routine SAVE aufgerufen werden. Wieder haben wir für Sie zur Verdeutlichung zwei Listings: Listing 11 zeigt, wie man ein Programm an eine beliebige Adresse lädt; Listing 12, wie man einen beliebigen Bereich auf Diskette speichert. Erwähnenswert ist noch die Routine CLALL, die alle Files im Computer schließt; die Kanäle in der 1541 bleiben davon jedoch unberührt. Hier müssen Sie also sorgfältig mit CLOSE arbeiten, da Sie sonst Daten verlieren können.

```
Prinzip des Ladens von Programmen.
LDX #$08    ;  Geräteadresse
LDY #$00    ;  Sekundäradresse für
               relativ laden
JSR FILPAR  ;  und setzen.
LDX # (Filename LO-Byte)
LDY # (Filename HI-Byte)
LDA # (Filename Länge)
JSR FILNAM
LDA #$00    ;  LOAD-Flag
LDX # (Startadresse LO-Byte)
LDY # (Startadresse HI-Byte)
JSR LOAD
RTS
```

Listing 11. Das Laden von Programmen an beliebige Adressen funktioniert so

```
Prinzip des Speicherns von
Bereichen.

LDX #$08    ;  Geräteadresse
JSR FILPAR  ;  setzen
LDX # (Filename LO-Byte)
LDY # (Filename HI-Byte)
LDA # (Filename Länge)
JSR FILNAM  ;  setzen
LDX # (Startadresse LO-Byte)
LDY # (Startadresse HI-Byte)
STX  $FB    ;  zwischenspeichern
STY  $FC
LDA #$FB    ;  Pointer auf Startadr.
LDX # (Endadresse +1 LO-Byte)
LDY # (Endadresse +1 HI-Byte)
JSR SAVE
RTS
```

Listing 12. Und so funktioniert das Speichern

Spooling? Was ist das?

Nachdem wir Sie nun mit Theorie überschwemmt haben, sollen Sie sogleich in den Genuß Ihrer neuen Kenntnisse kommen. Haben Sie schon einmal etwas von Spooling gehört? Nein? Macht nichts, wir werden uns mit dieser Technik nämlich jetzt auseinandersetzen, und Sie werden dabei die Vorzüge dieser Möglichkeit genießen lernen.

Unter dem Begriff Spooling verbirgt sich eigentlich eine ganz einfache Technik, die jedoch enorme Vorteile besitzt: Es handelt sich um das Drucken direkt von Diskette. Haben Sie nicht auch schon öfters versucht, ein meterlanges Listing auf Papier zu bringen und den Drucker dabei mit wütenden Blicken zu größerer Eile aufgefordert, weil Sie nämlich unter Zeitdruck standen und sich bei der Arbeit keine Verzögerung erlauben konnten? Dann ist Spooling genau das Richtige für Sie. Bei dieser Methode wird ein Listing, das ausgedruckt werden soll, auf Diskette gebracht. Danach starten Sie ein Spooling-Programm und siehe da; der Drucker beginnt, Ihr Listing auf Papier zu bringen, und der Computer meldet sich betriebsbereit mit READY.

Dies ist kein Wunder, sondern die Eigenschaft des seriellen Bus Ihres Computers. Sie haben vorhin gelernt, wie man den Bus des Computers in Maschinensprache bedient. Dabei fielen auch Worte wie TALK, LISTEN, SENDEN und EMPFANGEN. Der Trick des Spooling ist nun der: Mit Hilfe des CMD-Befehls in Basic können Sie ein Listing auf Diskette »umleiten« und zwar geschieht dies ähnlich wie beim Drucker: Sie öffnen ein File und schicken mit dem CMD-Kommando alle weiteren Bildschirmausgaben auf den Bus. Nur ist jetzt nicht der Drucker der Adressat, sondern die Floppystation. Hier ein Beispiel:
Sie haben ein Listing im Speicher und wollen dieses auf Diskette ablegen, sein Name soll »TEST« sein:

```
OPEN 1, 8, 2, "TEST,U,W"
CMD 1
LIST
```

Nach dieser Befehlsfolge wird Ihr Listing als USR-File auf Diskette geschrieben. Wie wäre es nun, wenn die Floppystation ein TALK-Kommando erhalten würde, das sie veranlaßt, das eben geschriebene File auf den Bus zu bringen? Der »Hörer« ist aber jetzt nicht, wie üblich, der Computer, sondern der Drucker, den wir zuvor mit einem LISTEN dazu aufgefordert haben. Die Folge wäre das, was Sie sich jetzt schon denken können:

Die Floppystation schickt das gesamte Listing über den Bus, und der Drucker, der ja auf Empfang programmiert ist, bekommt dieses Listing und druckt es aus. Der Computer hat mit der ganzen Sache nichts zu tun, da er sich nach Senden der Kommandos »zurückgezogen« hat und bleibt demzufolge frei für weitere Arbeit.

Drucken ohne Umwege

Der Zugriff auf den Bus ist dem Computer natürlich für die Zeit der Übertragung verwehrt, aber Sie können währenddessen intern weiterarbeiten. Ist die Übertragung beendet, so sind beide Peripheriegeräte noch auf Sendung und müssen erst »zur Ruhe gebracht« werden, bevor sie wieder ansprechbar sind. Aber auch das erledigt ein kleines Programm für uns. Sehen Sie sich jetzt einmal Listing 13 an. Es enthält ein Spooling-Programm, das mit

```
SYS 828, "filename"
```

aufgerufen wird. Danach meldet sich der Computer mit
SPOOLING filename
READY
und der Drucker beginnt zu arbeiten. Ist der Druckvorgang

```
-
.,  033C 20 79 00 JSR $0079
.,  033F F0 43    BEQ $0384
.,  0341 20 E7 FF JSR $FFE7
.,  0344 A0 00    LDY #$00'
.,  0346 B9 A5 03 LDA $03A5,Y
.,  0349 F0 06    BEQ $0351
.,  034B 20 D2 FF JSR $FFD2
.,  034E C8       INY
.,  034F D0 F5    BNE $0346
.,  0351 20 54 E2 JSR $E254
.,  0354 20 C1 F5 JSR $F5C1
.,  0357 A6 B7    LDX $B7
.,  0359 F0 66    BEQ $03C1
.,  035B A9 01    LDA #$01'
.,  035D A2 08    LDX #$08'
.,  035F A0 02    LDY #$02'
.,  0361 20 BA FF JSR $FFBA
.,  0364 20 C0 FF JSR $FFC0
.,  0367 A9 04    LDA #$04'
.,  0369 20 B1 FF JSR $FFB1
.,  036C 20 BE ED JSR $EDBE
.,  036F A2 01    LDX #$01'
.,  0371 20 C6 FF JSR $FFC6
.,  0374 20 BE ED JSR $EDBE
.,  0377 20 85 EE JSR $EE85
.,  037A 20 97 EE JSR $EE97
.,  037D A9 00    LDA #$00'
.,  037F 85 99    STA $99
.,  0381 85 98    STA $98
.,  0383 60       RTS
.,  0384 A9 01    LDA #$01'
.,  0386 85 98    STA $98
.,  0388 20 AE FF JSR $FFAE
.,  038B 20 AB FF JSR $FFAB
.,  038E A9 01    LDA #$01'
.,  0390 20 C3 FF JSR $FFC3
.,  0393 A0 00    LDY #$00'
.,  0395 B9 AF 03 LDA $03AF,Y
.,  0398 F0 06    BEQ $03A0
.,  039A 20 D2 FF JSR $FFD2
.,  039D C8       INY
.,  039E D0 F5    BNE $0395
.,  03A0 A9 00    LDA #$00'
.,  03A2 4C 74 A4 JMP $A474
.
.
.
.'03A5 53 50 4F 4F 4C 49 4E 47
.'03AD 20 00 45 4E 44 20 4F 46
.'03B5 20 53 50 4F 4F 4C 49 4E
.'03BD 47 8D 00 00 4C 08 AF 00
.
.
.
.,  03C1 4C 08 AF JMP $AF08
.
.
.
READY.
```

Listing 13. Mit diesem Programm können Sie ein Floppy-Drucker-Spooling durchführen. Näheres im Text.

beendet, so tippen Sie noch einmal
SYS828
ohne Filenamen, und die Leuchtdiode an der Floppy erlischt. Es erscheint die Meldung
END OF SPOOLING
READY
Dieses Programm ist, im Gegensatz zu unseren anderen Listings, zum sofortigen Eintippen gedacht.

Wir mußten jedoch leider feststellen, daß das Spooling nicht mit allen Druckern funktioniert. Bitte probieren Sie das von Fall zu Fall aus.

Wie Sie aus diesem Beispiel sehen, kann es von großem Nutzen sein, wenn Sie das Prinzip des seriellen Bus verstehen und dessen »Verkehrsregeln« kennen, da viele Programme nur deshalb mit geringem Aufwand große Effekte und Nutzen erzielen. Ein weiteres Beispiel in dieser Reihe dürfte wohl Hypra-Load sein, das Sie in Ausgabe 10/1984 des 64'er-Magazins fanden. Dieses Programm nutzt aber noch einige weitere Tricks der Maschinenspracheprogrammierung, die wir später besprechen wollen. Im folgenden wollen wir nämlich in die direkte Programmierung der Floppystation einsteigen, das heißt, das Speichern von Maschinenprogrammen in ihren Pufferspeicher und das Ausführen derselben mittels spezieller Befehle.

Das Aufzeichnungsformat

Zuerst wollen wir uns mit dem Aufzeichnungsformat der Diskette beschäftigen: Für einen einwandfreien Betrieb der Floppystation ist es unumgänglich, daß sich Markierungen auf der Diskette befinden. Diese Markierungen braucht das Laufwerk, um bestimmte Daten schnell finden zu können. Hierfür gibt es prinzipiell zwei Möglichkeiten: die Hardsektorierung und die Softsektorierung.

Hardsektorierte Disketten erkennt man daran, daß diese eine ganze Anzahl von Indexlöchern besitzen. Damit sind die kleinen Löcher nahe am Innenrand der Magnetscheibe gemeint. Mit einer Fotozelle können nun diese Löcher abgetastet werden, um die jeweilige Position der Diskette festzustellen. Dieses Verfahren hat den Vorteil, daß die Diskettenkapazität voll ausgenutzt werden kann. Es können so bis zu 5 MByte Daten auf eine 5¼-Zoll-Diskette geschrieben werden. Allerdings erfordert diese Methode einen enormen Hardware-Aufwand, der den Preis in die Höhe schnellen läßt. Für preiswerte Laufwerke (wie die 1541) geht man daher einen anderen Weg: die Softsektorierung. Hier besitzt die Diskette nur ein Indexloch zur Erkennung des Trackbeginns. Bei der 1541 ist sogar noch nicht einmal dieses erforderlich. Die notwendigen Markierungen werden beim Formatierungsvorgang softwaremäßig auf die Diskette gebracht, wobei natürlich wertvoller Speicherplatz verlorengeht. Softsektorierte Disketten im 5¼-Zoll-Format verfügen daher über zur Zeit maximal 1 MByte Speicherkapazität.

Uns soll also im weiteren die Softsektorierung beschäftigen, wobei in Bild 3 eine Diskette schematisch dargestellt ist, nachdem sie auf der 1541 formatiert wurde. Sie ist in 35 konzentrische Spuren, nachfolgend Tracks genannt, aufgeteilt. Jeder dieser Tracks enthält wiederum eine bestimmte Anzahl von Sektoren, die von außen nach innen abnimmt. Diese Tatsachen sind Ihnen aber schon bekannt. Nun wollen wir genauer auf den Aufbau der Sektoren einer Diskette eingehen.

Jeder Sektor besteht aus einem Blockheader und dem dazugehörigen Datenblock; eine schematische Darstellung zeigt Bild 4. Angeführt werden die Sektoren einer Diskette von den schon erwähnten Markierungen, die der Orientie-

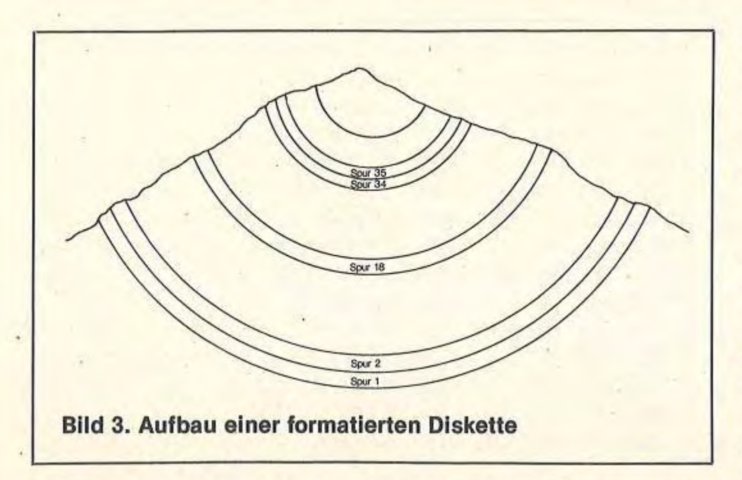


Bild 3. Aufbau einer formatierten Diskette

rung dienen. Diese Marken bezeichnet man als Synchron (SYNC)-Markierungen, sie bestehen aus mehreren $FF auf der Diskette. Erkennt der Schreib-/Lesekopf der Floppy also eine solche Marke, dann »weiß« die Floppystation, daß entweder ein Blockheader oder ein Datenblock nachfolgt. Nun müssen wir nur noch diese beiden voneinander unterscheiden können.

Hierzu dient das nächste Kennzeichen auf Diskette. Es folgt direkt nach der SYNC-Markierung und meldet dem Diskcontroller (DC), ob ein Blockheader oder ein Datenblock vorliegt. Hat das Kennzeichen den Wert $08, so handelt es sich um einen Blockheader; findet der Kopf hingegen den Wert $07, so handelt es sich um den Beginn eines Datenblocks.

Wir nehmen jetzt einmal an, der DC hätte das Kennzeichen $08 entdeckt; es handelt sich also um den Header eines Datenblocks. Dann folgt als nächstes Byte die Prüfsumme über den Header, die zur Kontrolle auf Lesefehler dient. Die Reihenfolge der Header-Bytes, wie sie im Commodore-Handbuch angegeben ist, stimmt nicht mit der Aufzeichnung auf Diskette überein.

Die nächsten zwei Byte stellen Sektor- und Track-Nummer dieses Sektors dar. Anhand dieser Werte kann der DC bei Track-Wechsel sehr schnell die Position des Schreib-/Lesekopfes ausfindig machen.

Das 5. und 6. Byte des Blockheaders gibt jeweils einen Teil der ID der Diskette an, und zwar folgen zuerst das zweite und dann das erste Zeichen der ID, die beim Formatieren festgelegt wurden. Mit diesen Angaben ist die Behandlung des Headers bereits abgeschlossen. Es folgen jetzt noch ein paar Bytes, die eine Lücke darstellen.

Mit der nächsten SYNC-Markierung wird der Beginn des eigentlichen Datenblocks eingeleitet. Nach der SYNC-Marke folgt das Datenblockkennzeichen $07. Die nächsten zwei Byte sind uns bestens bekannt. Sie können mit jedem Diskmonitor angesehen werden und geben Track- und Sektornummer des nächsten Blocks im File an. Man bezeichnet sie deshalb als Linker oder Link-Adressen (engl.: to link = verbinden).

Nun erst folgen die eigentlichen Daten auf Diskette, die in jedem Block 254 Byte ausmachen.

Hinter diesen Daten-Bytes steht die Prüfsumme des Datenblocks, die wiederum zum Erkennen von eventuellen Lesefehlern dient. Werden solche Fehler festgestellt, so versucht die Floppystation noch mehrere Male, den Block doch zu lesen. Erst wenn viele Versuche kein befriedigendes Ergebnis bringen, steigt sie mit einer Fehlermeldung aus.

Nach der Prüfsumme des Datenblocks folgt wieder eine Lücke auf der Diskette, bevor die SYNC-Markierung des nächsten Blockheaders kommt. Wenn wir uns diesen Aufbau eines Sektors betrachten, wird klar, warum die Speicherkapazität bei softsektorierten Disketten gegenüber hardsektorierten Disketten deutlich abnimmt.

Jetzt werden Sie vielleicht auch die Beschreibung der Fehlermeldungen im Floppy-Handbuch verstehen, die wir hier nicht mehr aufführen, da sie dort sehr genau und richtig erläutert werden.

Das Verständnis des Diskettenaufbaus ist für die weitere Behandlung des DOS unerläßlich, da wir nur so die Funktionsweise begreifen lernen.

Arbeitsweise des DOS

Jetzt wollen wir uns aber einmal mit der grundlegenden Arbeitsweise des Floppy-Betriebssystems (DOS) befassen, das um einiges komplizierter ist, als das des Computers.

Wenn wir das Laufwerk einschalten, passiert zunächst das gleiche wie im Computer. Die RESET-Leitung geht auf Low und der Mikroprozessor, hier ein 6502, holt sich seine Systemstartadresse. Danach läuft das RESET-Programm an, wobei die Floppystation einen Selbsttest durchführt. Erkennen können Sie dies daran, daß für kurze Zeit der Motor anläuft und die rote LED leuchtet. Wurde kein Defekt registriert, so erlischt die Leuchtdiode wieder, und der Motor geht aus. Jetzt wird der RAM-Bereich der Floppy initialisiert und alle wichtigen Zeiger werden hergestellt. Danach ist die 1541 betriebsbereit.

Von jetzt an laufen quasi drei Programme gleichzeitig ab:
- das Hauptprogramm läuft in einer Schleife, die nur bei der Ausführung von Befehlen verlassen wird;
- das Diskcontroller-Programm wird über den IRQ gesteuert und durch den Timer des Diskcontrollers (DC) alle 10 ms aufgerufen;
- die Routinen des Buscontrollers (BC) schließlich werden nur im Bedarfsfall aufgerufen, nämlich, wenn die ATN-Leitung des seriellen Bus auf Low geht.

Wir wollen uns die Funktion dieser Routinen nun einmal etwas genauer betrachten.

Das Hauptprogramm

Das Hauptprogramm hängt, wie schon gesagt, in einer Warteschleife, bis ein Befehl vom Computer kommt. Dieser aktiviert zuerst die Busroutinen, die die gesendeten Bytes dann entgegennehmen und speichern. Jetzt bekommt das Hauptprogramm, das übrigens den Zustand der beiden Steuer-Routinen (DC und BC) ständig überwacht, die Meldung, daß ein Befehl anliegt. Es verzweigt nun zur Befehls-

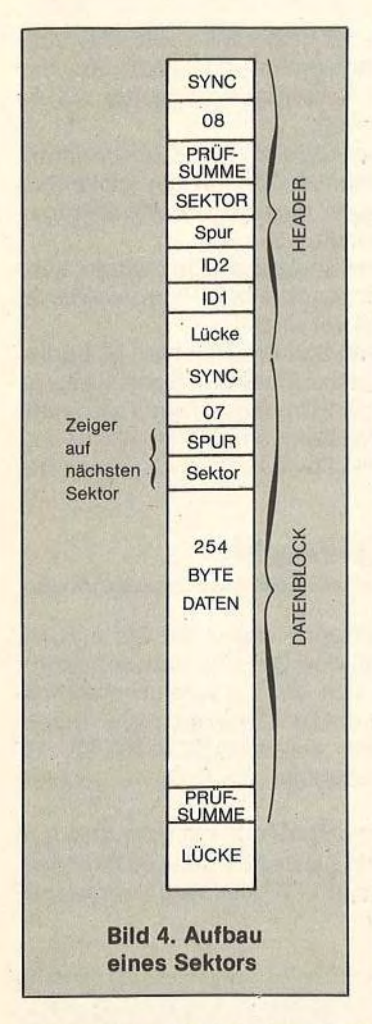


Bild 4. Aufbau eines Sektors

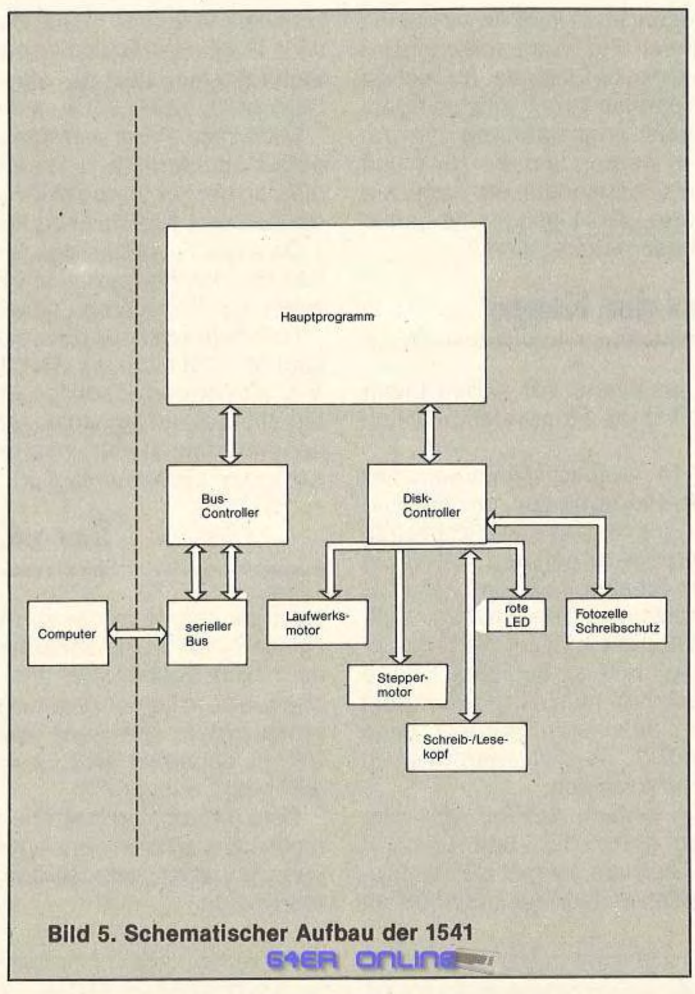


Bild 5. Schematischer Aufbau der 1541

auswertung, ähnlich dem Basic-Interpreter, und führt gegebenenfalls einen Befehl aus, sofern kein Syntaxfehler entdeckt wurde. In diesem Fall würde sonst eine Fehlermeldung generiert, die dann vom Computer ausgelesen werden kann.

Ist ein Befehl korrekt ausgeführt worden, so werden die Befehlsparameter wieder gelöscht, und das Hauptprogramm kehrt in die Warteschleife zurück.

Das Diskcontroller-Programm

Der Diskcontroller enthält den Baustein VIA 6522, durch den er mit dem Mikroprozessor in Kontakt steht. Dieser Baustein enthält auch Timer, die in einem eingestellten Rhythmus einen IRQ auslösen können. Einer dieser Timer ist in der 1541 so eingestellt, daß er ungefähr alle 10 ms einen IRQ auslöst, der dann seinerseits das Diskcontroller-Programm aufruft.

Es soll an dieser Stelle der Unterschied zwischen Diskcontroller und Diskcontroller-Programm erläutert werden: Als Diskcontroller (DC) bezeichnet man die Hardware in der Floppystation, die für den Laufwerksbetrieb zuständig ist.

Unter dem Diskcontroller-Programm versteht man den Programmteil im DOS, der, durch IRQ geregelt, die Ansteuerung des DC übernimmt.

Eine vollständige Trennung dieser beiden Begriffe ist jedoch weder notwendig noch zweckmäßig, so daß wir mit dem Ausdruck »DC« immer die Gesamtheit von Hard- und Software beschreiben wollen. Nun aber wieder zu den Aufgaben des DC.

Auch dieses Programm hat eine Art Wartezustand, solange kein Befehl vom Computer anliegt. Wird nämlich das Hauptprogramm über den Bus aktiviert, so wertet dieses die Befehle aus und gibt sie an den DC weiter, der dann seinerseits dafür sorgt, daß das Laufwerk aktiviert wird. Er steuert den Laufwerk- und den Stepper(Schreib-/Lesekopf)-Motor und bedient die Daten, die vom und zum Tonkopf gehen.

Die gesamten Vorgänge am Laufwerk werden also interruptgesteuert vorgenommen.

Die Busroutinen

Die Routinen des Buscontrollers (BC) werden ebenfalls über die IRQ-Leitung gesteuert. Auch der BC enthält einen VIA 6522-Baustein. Hier wird der Aufruf der Routinen allerdings nicht über den Timer organisiert, sondern, wie schon erwähnt, über die ATN-Leitung des seriellen Bus. Zieht der Computer also diese Leitung auf Low, so wird in der Floppystation (und in allen anderen Peripheriegeräten ebenso) ein IRQ ausgelöst. Dann erfolgt die Abfrage, ob dieser IRQ vom Timer des DC kam. Ist dies nicht der Fall, so wird die BC-Routine aufgerufen, die dann den weiteren Busbetrieb übernimmt. Sollte die Floppystation gerade einen Befehl bearbeiten, während schon ein neuer vom Computer gesendet wird, so wartet der BC so lange mit der Annahme, bis das Laufwerk wieder in den Bereitschaftszustand zurückgekehrt ist.

Wie Sie sehen, stellt das DOS eine ziemlich komplizierte Einheit dar, deren Schema in Bild 5 zu sehen ist.

Wollen wir also in dieses System einsteigen, um dort eigene Programme ausführen zu lassen, so ist es natürlich unerläßlich, daß wir die »Spielregeln« dieses Prozessorsystems genau kennen, da es sonst leicht zu kleinen Katastrophen kommen kann.

Disketten werden auf ihrer Rückseite beschrieben!

Zu Ihrer weiteren Arbeit mit der 1541/70/71 noch ein paar Tips: Wenn Sie vorhaben, Programme im Floppy-RAM ablaufen zu lassen, sollten Sie Ihre Floppystation öffnen und ohne Deckel betreiben. So können Sie genau beobachten, wie der Kopf positioniert wird und was bei Lesefehlern geschieht. Sie werden unter anderem auch entdecken, daß Disketten nicht etwa auf der Seite beschrieben werden, auf der sich das Etikett befindet, sondern auf der Rückseite. Dies ist um so bemerkenswerter, als man eine Diskette immer nur auf der Vorderseite schonend behandelt, die ja eigentlich nicht benutzt wird. Auch wir mußten die Erfahrung machen, daß wir Disketten lange Zeit mit der wertvollen Seite auf Tische gelegt haben, stets darauf achtend, daß ja kein Staubkorn auf die von uns so gehütete Vorderseite kam.

Das Betreiben ohne Deckel hat auch den Vorteil besserer Wärmeableitung. Die Mechanik wird es Ihnen danken.

Nachdem Sie Ihre 1541 also auf »Arbeitsbetrieb« getrimmt haben, wollen wir gleich einmal mit kleinen Programmen beginnen. In Tabelle 11 sehen Sie eine Aufstellung einiger wichtiger Zero-Page-Adressen, die uns im weiteren Verlauf noch beschäftigen werden. Für ein DOS-Listing ist in unserer

Serie natürlich kein Platz vorhanden; auch können wir nur mit kleinen Beispielen versuchen, Ihnen die Programmierung der 1541 nahezubringen. Für diejenigen unter Ihnen, die jedoch vorhaben, tiefer in die Floppy-Programmierung einzusteigen, seien an dieser Stelle zwei Bücher angesprochen, die von Markt & Technik herausgegeben werden und die 1541 und die 1570/71 bis ins kleinste Detail behandeln. Sie enthalten außerdem sehr gut dokumentierte DOS-Listings und gehen weit über das in diesem Kurs besprochene hinaus.

Programmieren der Floppy

So, jetzt soll es aber endlich losgehen. Wir wollen unser erstes Programm schreiben und in der Floppystation ablaufen lassen.

Es handelt sich um Listing 14. Dieses »Miniprogramm« schreiben wir in den Puffer 0 der Floppystation, das heißt ab Adresse $0300. Das Basic-Programm haben wir der Kürze halber gleich an den Assemblercode angehängt. Wenn Sie das Programm starten, wird das Bit abgefragt, das beim DC für den Zustand der Schreibschutzplakette verantwortlich ist. Sie werden vielleicht wissen, daß das Laufwerk die Schreibschutzkerbe bei den Disketten mit Hilfe einer Lichtschranke abfragt. Ist die Lichtschranke unterbrochen, das heißt es liegt eine Diskette mit Schreibschutzaufkleber im Laufwerk, dann steht das entsprechende Bit auf 0. Tesafilm kann deshalb nicht als Schreibschutz verwendet werden.

Unser Programm schiebt nun einfach das Bit der Lichtschranke an die Stelle des Bits für die rote LED und speichert diesen Wert wieder ab. Starten Sie einmal unser kleines Programm, dann werden Sie feststellen, daß die Leuchtdiode am Laufwerk erlischt, wenn die Lichtschranke unterbrochen wird. Holen Sie die Diskette dagegen aus dem Laufwerk oder legen Sie eine Diskette ohne Schreibschutzplakette ein, so beginnt die rote LED zu leuchten.

Mit diesem Programm können Sie also testen, ob von Ihnen selbst angefertigte Schreibschutzkerben in der Diskettenhülle an der richtigen Stelle liegen, um eine Diskette eventuell doppelseitig benutzen zu können.

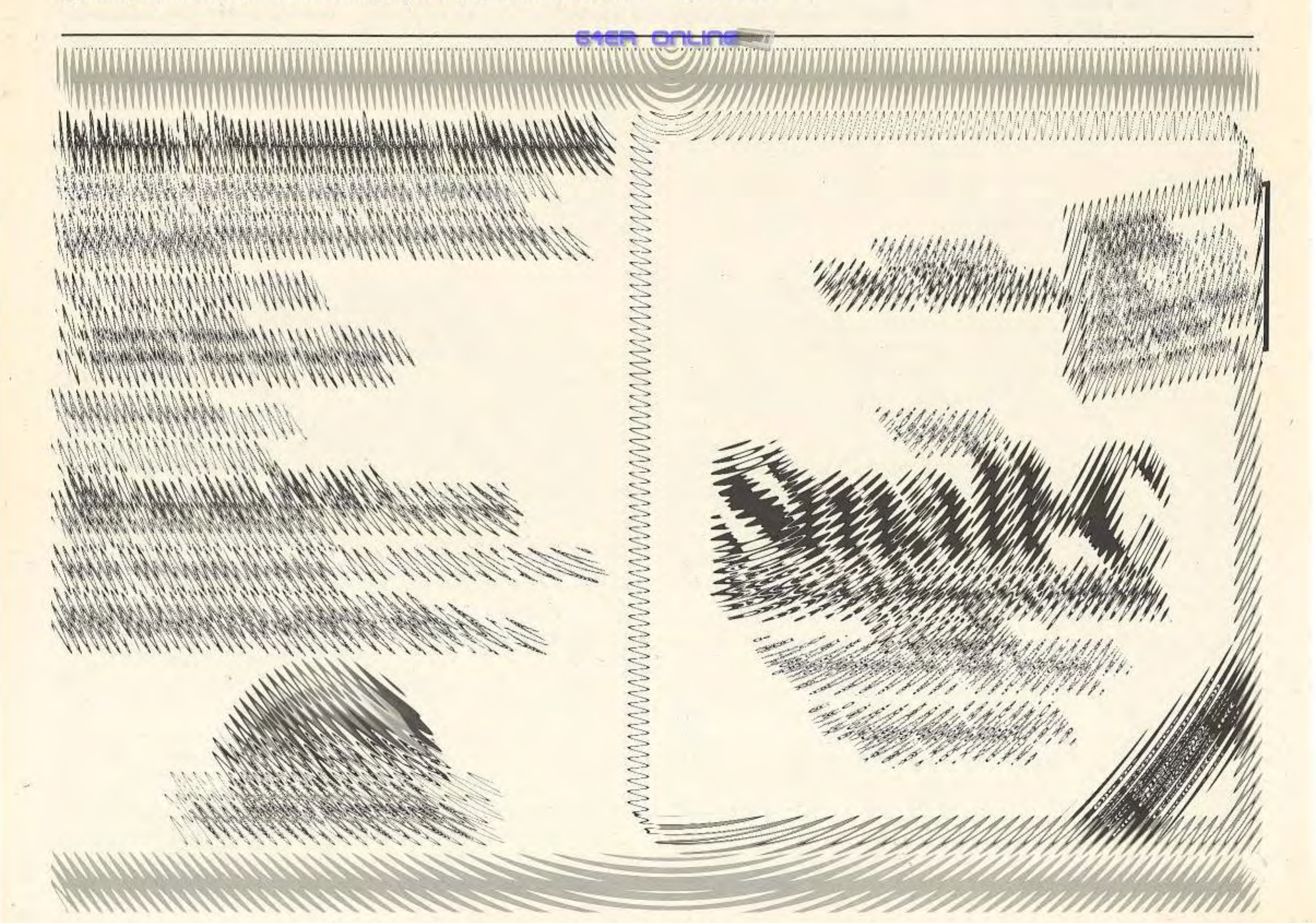
Da unser Programm aus einer Endlosschleife besteht, können Sie die Floppystation nur durch einen Reset wieder in einen ansprechbaren Zustand versetzen.

Das Programm hat aber einen Schönheitsfehler; es beeinflußt nämlich nicht nur die beiden LED-Bits in Speicherstelle $1C00, sondern löscht bei jedem Durchgang auch alle anderen Bits dieses Registers, deren Belegung Sie Tabelle 12 entnehmen können. Für unsere Testzwecke ist diese »Pfuscherei« jedoch unwesentlich.

Der »&«-Befehl

Nach diesem aufregenden Beispiel wollen wir Sie nun mit einem Befehl bekanntmachen, den Sie sehr wahrscheinlich noch nicht kennen. Er nennt sich »&« und wird unverständlicherweise in keiner Anleitung gut beschrieben. Der &-Befehl entspricht in gewisser Weise einem BLOCK-EXECUTE-Befehl; auch hier wird ein Programm von Diskette geladen und sofort ausgeführt.

Der Unterschied besteht nur darin, daß mit dem &-Befehl nicht nur ein Block, sondern ein ganzes File, das im Directory verzeichnet ist, geladen und im Puffer als Programm ausgeführt wird.



$0000	Jobspeicher für Puffer 0
$0001	Jobspeicher für Puffer 1
$0002	Jobspeicher für Puffer 2
$0003	Jobspeicher für Puffer 3
$0004	Jobspeicher für Puffer 4
$0005	Jobspeicher für Puffer 5 (im RAM nicht vorhanden)
$0006/7	Spur- und Sektornummer für Befehl in Puffer 0
$0008/9	Spur- und Sektornummer für Befehl in Puffer 1
$000A/B	Spur- und Sektornummer für Befehl in Puffer 2
$000C/D	Spur- und Sektornummer für Befehl in Puffer 3
$000E/F	Spur- und Sektornummer für Befehl in Puffer 4
$0010/1	Spur- und Sektornummer für Befehl in Puffer 5
$0012/3	ID der Diskette im ASCII-Code; die beiden Zeichen der aktuellen ID werden bei jedem Blocksuchbefehl gelesen und hier aktualisiert abgespeichert. Auch das Initialisierkommando benutzt diesen Befehl und bringt die ID dadurch auf den neuesten Stand.
$0016- $001A	Hier sind die Bytes für den aktuellen Blockheader gespeichert, und zwar sind dies: $0016 erstes Zeichen der ID $0017 zweites Zeichen der ID $0018 Spurnummer des Blocks $0019 Sektornummer des Blocks $001A Prüfsumme über den Blockheader Auf der Diskette stehen diese Werte in der umgekehrten Reihenfolge!
$001C	Flag für Änderung beim Schreibschutz der Diskette
$002E/F	Zwischenspeicher für aktuelle Zeiger
$0030/1	Zeiger in aktuellen Puffer
$0032/3	Zeiger auf aktuellen Blockheader beim Schreiben
$0038	Kennzeichen ($07) für Beginn eines Datenblocks
$0039	Kennzeichen ($08) für Beginn eines Blockheaders
$003A	Zwischenspeicher für Prüfsummen
$003D	aktuelle Laufwerksnummer; bei der VC 1541 immer 0
$003E	gerade arbeitendes Laufwerk ($FF = kein Laufwerk)
$003F	Puffernummer des eben ausgeführten Befehls (0-5)
$0043	zählt die Anzahl der Sektoren bei der Formatierung
$0044	Zwischenspeicher beim Arbeiten
$0045	Zwischenspeicher für aktuellen Befehlscode
$0047	enthält aktuelles Kennzeichen für Beginn eines Datenblocks. wird nur bei RESET einmal auf $07 gesetzt und kann vom Benutzer verändert werden, wobei das Hi-Nybble des Wertes immer auf 0 ($0-) stehen sollte, um Leseprobleme des DC zu vermeiden. Wird versucht, einen Datenblock mit einer anderen, als der hier gespeicherten, Nummer zu lesen, so erfolgt der Fehlercode $04 des DC und die Floppy sendet Fehlermeldung Nummer 22 zum Bus.
$0049	Zwischenspeicher für den Stackpointer
$004A	Zähler für Kopftransport; Zahlen bis 127 bewegen den Kopf nach außen; Zahlen von 128 bis 255 bewegen ihn nach innen (höhere Spurnummer).
$0051	aktuelle Spurnummer bei der Formatierung; steht auf $FF, wenn keine Formatierung erfolgt.
$0065/6	Zeiger auf die NMI-Routine; wird bei einem RESET gestellt.
$0067	Flag zum Anzeigen eines NMI
$0068	Flag zum Ermöglichen (0) oder Sperren (1) der automatischen Initialisierung einer Diskette, falls ein ID Type Mismatch Error erkannt wurde
$0069	Abstand der Sektoren bei der Zuteilung; erhält bei einem RESET den Wert 10.
$006A	Anzahl der Leseversuche eines Sektors; steht nach RESET auf 5.
$006B/C	Zeiger auf Sprungtabelle der USER-Befehle; steht normalerweise auf $FFF6 nach einem RESET.
$006D/E	Zeiger auf den Beginn der 'Bit Map'; steht auf $0400 und wird beim Initialisieren gesetzt.
$006F	Zwischenspeicher; steht nach RESET auf $6F
$0070	Zwischenspeicher
$0071	Zwischenspeicher
$0072	Zwischenspeicher; steht nach RESET auf $FF
$0073	Zwischenspeicher
$0074	Zwischenspeicher
$0075/6	Indirekter Zeiger auf $0100; wird bei RESET gestellt
$0077	Gerätenummer + $20 für das LISTEN-Kommando
$0078	Gerätenummer + $40 für das TALK-Kommando
$0079	Flag für LISTEN (1/0)
$007A	Flag für TALK (1/0)
$007B	Flag für Adressierung
$007C	Flag für ATN-Signal vom seriellen Bus
$007D	Flag für Prozessor im ATN-Modus
$007F	Aktuelle Laufwerksnummer; hier immer 0
$0080	Aktuelle Spurnummer; enthält $00 nach Ausführung
$0081	Aktuelle Sektornummer; enthält $00 nach Ausführung
$0082	Aktuelle Kanalnummer
$0083	Aktuelle Sekundäradresse
$0084	übliche Sekundäradresse
$0085	Aktuelles Datenbyte
$0086	Speicher für Zwischenergebnisse
$0087	Speicher für Zwischenergebnisse
$0088	Speicher für Zwischenergebnisse
$0089	Speicher für Zwischenergebnisse
$008A	Speicher für Zwischenergebnisse
$008B- $008E	Speicher für Ergebnisse bei Berechnungen
$008F- $0093	Akkumulator für Berechnungen
$0094/5	Zeiger auf Directory-Puffer; enthält $05/02
$0096	Kommando vom IEEE-Bus; hier unbenutzt
$0098	Bitzähler für seriellen Bus
$0099/A	Buffer-Pointer für Puffer 0; steht auf $0300
$009B/C	Buffer-Pointer für Puffer 1; steht auf $0400
$009D/E	Buffer-Pointer für Puffer 2; steht auf $0500
$009F/0	Buffer-Pointer für Puffer 3; steht auf $0600
$00A1/2	Buffer-Pointer für Puffer 4; steht auf $0700 Alle diese Pointer werden durch den B-P-Befehl verändert!
$00A3/4	Zeiger auf nächstes Zeichen im INPUT-BUFFER ($0200)
$00A5/6	Zeiger auf nächstes Zeichen im ERROR-BUFFER ($02D6)
$00A7- $00AD	Tabelle; enthält für jeden aktiven Puffer die entsprechende Kanalnummer. Kanalnummer = $FF, wenn Puffer unbenutzt.
$00AE- $00B4	Tabelle; enthält für jeden aktiven Puffer die entsprechende Kanalnummer. Kanalnummer = $FF, wenn Puffer unbenutzt.
$00B5- $00BA	Tabelle der Lo-Bytes der Recordnummern für jeden Puffer
$00BB- $00C0	Tabelle der Hi-Bytes der Recordnummern für jeden Puffer
$00C1- $00C6	Tabelle der nächsten zu bearbeitenden Recordnummern für jeden Puffer
$00C7- $00CC	Tabelle der Recordlängen für jeden Puffer
$00CD- $00D2	Tabelle der Side-Sektoren für jeden Puffer
$00E2- $00E6	Standardwerte für Laufwerk; hier alle 0
$00E7- $00EB	Tabelle der Filetypen
$00EC- $00F1	Kanal Filetyp
$00F2- $00F7	Kanalstatus
$00F8	Zwischenspeicher für EOI
$00F9	Aktuelle Puffernummer für Befehlscode
$0101	Formatkennzeichen von Spur 18 Sektor 0
$0104- $0145	Bereich des Hardware-Stack; nicht benutzbar
$0200- $0229	INPUT-BUFFER; hier werden alle Befehlsstrings vom Computer zwischengespeichert und nach Syntaxprüfung aus geführt
$022A	Codenummer des auszuführenden Befehls
$022B-	Kanaltabelle; diese Tabelle enthält für jede mögliche
$023E-	Aktuelles Datenbyte für jeden Kanal; Belegung der
$0243	Adressen wie bei der Kanalstatustabelle ($022B)
$0244- $0249	Tabelle der Zeiger auf das letzte aktuelle Zeichen in jedem, für den Kanal zuständigen, Pufferspeicher
$024A	Gerade behandelter Filetyp
$024B	Länge des Befehlsstrings
$024C	Zwischenspeicher für Sekundäradresse
$024D	Zwischenspeicher für Befehlscode
$024E	Arbeitsspeicher beim Suchen des nächsten Sektors
$024F/0	Pufferbelegungsspeicher; 1 = Puffer belegt
$0253	Flag für Directory-Eintrag gefunden
$0254	Flag für $-Befehl zum Listen des Directory
$0255	Flag für Befehlsausführung (<>$00, wenn Befehl anliegt)
$0257	Nummer des letzten benutzten Puffers
$0258	Recordlänge
$0259	Side-Sector Spur
$025A	Side-Sector Sektor
$025B- $025F	Tabelle; enthält den letzten Befehlscode der Puffer
$0260- $0265	Sektornummern der Directoryeinträge in den Puffern
$0266- $026B	Zeiger auf die Directoryeinträge in den Puffern
$026D	Flag für LED Blinken bei Fehler
$026E	Nummer des letzten aktiven Laufwerks
$026F	Nummer des letzten bearbeitenden Sektors
$0270	aktueller Schreibkanal
$0271	aktueller Lesekanal
$0274	Länge des Befehlsstrings im INPUT-BUFFER
$027A- $027F	Tabelle der Zeiger auf die Filenamen
$0280- $0284	Spurnummern der Files für den aktuellen Puffer
$0285- $0289	Sektornummern der Files für den aktuellen Puffer
$028A	Joker (*) Flag
$028E	Standardwert für die Nummer des Laufwerks
$028F	Flag für Fileeintrag im Directory gefunden
$0290	Sektornummer des aktuellen Directory Sektors
$0291	Sektornummer des ersten Directoryeintrags
$0292	Zeiger auf ersten gültigen Directoryeintrag
$0293	Zeigt letzten Block an; enthält dann 0
$0294	Aktueller Pufferzeiger
$0295	Zähler für Fileeinträge
$0297	Betriebsart des aktuellen Files (Lesen/Schreiben)
$029D/E	Spurnummer der BAM
$02A1- $02B0	Zwischenspeicher für BAM Eintragungen
$02B1- $02D4	Puffer für Directory
$02D5- $02F8	ERROR-BUFFER; enthält auszugebende Fehlermeldung
$02FA	Lo-Byte der Anzahl der freien Blocks auf Diskette
$02FC	Hi-Byte der Anzahl der freien Blocks auf Diskette
$0300- $03FF	Puffer 0
$0400- $04FF	Puffer 1
$0500- $05FF	Puffer 2
$0600- $06FF	Puffer 3
$0700- $07FF	Puffer 4 (enthält normalerweise die BAM)
$0800- $FFFF	Nicht mit RAM belegt

Tabelle 11. Die wichtigsten Zero-Page-Adressen der Floppy

Außerdem müssen die Files, die mit dem Befehl »&« gestartet werden sollen, speziell gekennzeichnet sein. Sie enthalten als erstes Zeichen im Filenamen das Zeichen »&«. Soll also zum Beispiel ein File mit dem Namen »Test« als Autostartprogramm in der 1541 ausgeführt werden, so geben Sie diesem File den Namen »&Test« und starten Sie es danach mit:
`OPEN1,8,15,"&TEST"`

Haben Sie nur ein einziges Autostartfile auf Diskette, so können Sie es auch nur mit »&« speichern und ebenso mit `OPEN 1,8,15," &"` starten

Leider erwartet die Floppystation von Autostart-Files eine spezielle Syntax, die in Tabelle 13 zu sehen ist.

Als Listing 15 haben wir noch einmal unser LED-Testprogramm; nur wird diese Routine durch das Basic-Programm als &-File auf Diskette geschrieben, und kann danach durch den schon erwähnten Befehl direkt von Diskette in den Pufferspeicher geschrieben und dort gestartet werden.

Zu Tabelle 13 noch einige Anmerkungen:

Zuerst muß die Startadresse des Programms im Pufferspeicher der 1541 in das File geschrieben werden. Danach folgt die Anzahl der Bytes im Programm. Jetzt werden die Programmbytes gespeichert, und schließlich folgt noch eine Prüfsumme, die sich wie folgt errechnet:

Komplizierte Prüfsummenberechnung

Es werden alle Bytes des Programms addiert und zum Ergebnis noch die zwei Byte der Startadresse und die Anzahl der Bytes im Programm hinzugezählt. Dieses Ergebnis ist als Integerzahl zu verstehen und besteht also aus einem niederwertigen (LO) und einem höherwertigen (HI) Byte. Das niederwertige Byte ist die Prüfsumme, zu der noch der Übertrag im höherwertigen Byte addiert werden muß. Diese Berechnung klingt kompliziert; ist es aber nicht. Die allgemeine Formel hier noch einmal:

HB=INT(SUMME/256)
LB=SUMME-HBx256

dabei bedeuten:

HB - das höherwertige Byte
LB - das niederwertige Byte
SUMME - die Gesamtsumme der Programm-Bytes

Achtung: Die Übertragsberechnung muß nach jedem neu dazugezählten Wert erfolgen, da das Endergebnis kleiner als 256 sein muß! Wie Sie sehen, ist das Anlegen eines &-Files nicht ganz einfach. Bisher wurde diese Fileart fast nur von Profis zum Programmschutz angewandt, da sie, wie schon erwähnt, nahezu unbekannt war.

Zu erwähnen wären noch zwei seltsame Fehlermeldungen der 1541:

»OVERFLOW IN RECORD« erscheint, wenn die Anzahl der tatsächlichen Bytes mit der Angabe nicht übereinstimmt.

»RECORD NOT PRESENT« erscheint, wenn die Prüfsumme nicht stimmt.

Da wir stets darum bemüht sind, Ihnen die Arbeit mit der Floppystation so angenehm wie möglich zu machen, haben wir unserem Artikel noch Listing 16 beigefügt. Es handelt sich hier um ein Programm, das es Ihnen gestattet, auf einfachste Weise &-Files zu erstellen. Diese können sogar länger als 256 Byte sein, da das Programm dann automatisch eine Prüfsumme und die Anschlußadresse einfügt. Ununterbrochene &-Files, die länger als 256 Zeichen sind, kann es ja nicht geben, da die Anzahl der Programm-Bytes im File nur in einem Byte gespeichert wird.

Die fortgeschrittenen Programmierer unter Ihnen werden sicher schon mit Ungeduld auf den Beginn des folgenden Abschnitts gewartet haben. Jetzt wird unser Kurs seinem Titel nämlich endlich voll gerecht werden, und wir wollen einmal sehen, was sich so alles mit einer Diskette anstellen läßt.

```
0 GOTO 10                                  <234>
1 , 0300 AD 00 1C LDA $1C00                <018>
2 , 0303 29 10    AND #$10                 <023>
3 , 0305 4A       LSR                      <093>
4 , 0306 8D 00 1C STA $1C00                <041>
5 , 0309 4C 00 03 JMP $0300                <005>
6 :                                          <064>
10 OPEN 1,8,15                             <208>
20 FOR X=0 TO 11:READ A                    <215>
30 PRINT#1,"M-W"CHR$(X)CHR$(3)CHR$(1)CHR$(A)
   :NEXT                                   <065>
40 PRINT#1,"M-E"CHR$(0)CHR$(3)             <179>
50 DATA 173,0.28,41,16,74,141,0,28,76,0,3  <005>
```

Listing 14. Unser erstes Floppy-Maschinenprogramm

DISKCONTROLLER (DC)

VIA 6522, $1800, PORT B

Bit #	Bedeutung
0	DATA IN
1	DATA OUT
2	CLOCK IN
3	CLOCK OUT
4	ATN OUT
5 6	GERÄTENUMMER
7	ATN IN (CB 2)

BUSCONTROLLER (BC)

VIA 6522, $1C00, PORT B

Bit #	Bedeutung
0	Steppermotor - Spule 1
1	Steppermotor - Spule 2
2	Laufwerksmotor
3	LED am Laufwerk (rot)
4	Schreibschutzkennung
5	Bitsynchronisation für DC bei den vier
6	Spurbereichen
7	SYNC-Signal

Tabelle 12. Belegung der beiden Control-Ports der 1541

Byte	Bedeutung
1-2	Startadresse in der 1541 im HI/LO-Format
3	Anzahl der folgenden Programmbytes
4-N	Programm
N+1	Prüfsumme
N+2	Hier kann bei längeren Programmen ein weiterer Teil eingefügt werden. Format: wieder bei Byte 1 beginnend.

Tabelle 13. Aufbau eines &-Files. In dieser Tabelle sind die Linker- beziehungsweise Endekennzeichen, die in den ersten beiden Byte eines Datenblocks stehen, nicht enthalten, da sie beim Öffnen und Beschreiben eines &-Files automatisch gesetzt werden.

Selbstverständlich sollen dabei Errors (Diskettenfehler) und »Killertracks« auch nicht zu kurz kommen.

Damit wir uns aber wieder an wichtige Tatsachen erinnern, noch einmal eine kurze Zusammenfassung einiger wichtiger Einzelheiten.

Wie ausführlich beschrieben, besteht ein Sektor auf Diskette aus zwei Teilen, nämlich dem Header und dem eigentlichen Datenblock. Beide Teile des Sektors werden auf Diskette durch eine SYNC-Markierung angekündigt, der dann

```
100 REM ERZEUGT &-FILE, DAS LISTING 2            <220>
110 REM (LED-TEST) ENTSPRICHT.                   <194>
120 :                                            <178>
130 DATA 0,7,12,173,0,28,41,16,74,141            <093>
140 DATA 0,28,76,0,7,93                          <197>
150 OPEN 1,8,2,"&,U,W"                           <194>
160 FOR X=1 TO 16:READ A                         <105>
170 PRINT#1,CHR$(A);:NEXT X                      <071>
180 CLOSE 1                                      <133>
```

Listing 15. So macht man Listing 14 zu einem »&-File«

```
100 REM AUTO-'&'-MAKER                           <106>
110 REM ---------------                          <115>
120 REM                                          <007>
130 REM  03.11.84.  BORIS SCHNEIDER              <224>
140 :                                            <198>
150 :                                            <208>
160 REM INITIALISIERUNG                          <159>
170 INPUT"STARTADRESSE DES &-FILES";SA           <121>
180 INPUT"NAME DES &-FILES";NA$                  <046>
190 IF LEN(NA$)>15 THEN 180NA$                   <026>
200 OPEN 1,8,2,"&"+NA$+",U,W"                    <063>
210 DIM X(256)                                   <158>
220 PRINT"BITTE GEBEN SIE JETZT IHRE DATEN EIN"
                          <116>
230 PRINT"ABSCHLUSS MIT -1!"                     <212>
240 :                                            <042>
250 REM DATENEINGABE UND TEST AUF                <227>
260 REM UEBERLAUF                                <047>
270 Y=1                                          <075>
280 INPUT X(Y)                                   <160>
290 IF X(Y)<0 THEN Y=Y-1:GOTO 350                <213>
300 PR=PR+X(Y):IF PR>255 THEN PR=PR-255          <103>
305 Y=Y+1:IF Y>254 THEN 350                      <026>
310 GOTO 280                                     <090>
320 :                                            <123>
330 REM ABSPEICHERN DER VORHANDENEN              <017>
340 REM DATEN IN DAS USR-FILE                    <006>
350 SH=INT(SA/256)                               <144>
360 SL=SA-256*SH                                 <221>
370 PR=PR+SH+SL+Y                                <250>
380 PRINT#1,CHR$(SL);CHR$(SH);                   <082>
390 PRINT#1,CHR$(Y);                             <040>
400 FOR I=1 TO Y                                 <059>
410 PRINT#1,CHR$(X(I));                          <213>
420 NEXT                                         <039>
430 PR=PR-(255*INT(PR/256))                      <219>
440 PRINT#1,CHR$(PR);                            <163>
450 IF X(Y+1)<0 THEN GOTO 470                    <217>
460 SA=SA+Y:PR=0:GOTO 270                        <196>
470 CLOSE 1                                      <168>
```

Listing 16. Komfortable »&-Files« erzeugen

Wichtige Adressen des DOS:

- $FD9E – Rücksprung in die Jobschleife
- $F556 – Sync-Signal auf Diskette abwarten
- $FE00 – PCR auf Lesen umschalten
- $FE0E – Track mit $55 vollschreiben
- $FDA3 – Track mit SYNC vollschreiben
- $F510 – Blockheader lesen:
  - + Diskette muß initialisiert sein
  - + $32/33 muß die Adresse der Track- und Sektornummer enthalten (L/H); zum Beispiel $00/03, wenn die Nummern in $0300/0301 abgespeichert sind
  - + Rückkehr nur bei fehlerfreier Durchführung des Lesens
- $F527 – Blockheader lesen:
  - + Diskette muß initialisiert sein
  - + zuvor muß $12 nach $16 und $13 nach $17 gebracht werden
  - + Track- und Sektornummer in $18 und $19
  - + Rückkehr nur bei fehlerfreier Durchführung des Lesens
- $F50A – Datenblockanfang suchen:
  - + Parameter siehe $F510

Tabelle 14. Einige Unterprogramme des DOS

das Kennzeichen (ob Header- oder Datenblock) zur Identifikation folgt.

Der Blockheader enthält Track- und Sektornummer des Blocks, die beiden Bytes der Diskettenidentifikation (ID) und schließlich noch eine Prüfsumme, die dem Diskcontroller (DC) mitteilt, ob alle Daten einwandfrei gelesen wurden. Wurde der Blockheader richtig eingelesen, so wartet der DC auf den nachfolgenden Datenblock, der die Zeiger auf den nächsten Block im File, die Daten-Bytes und schließlich ebenfalls eine Prüfsumme enthält.

Zwischen Blockheader und Datenblock und zwischen Datenblock und Header des darauffolgenden Sektors befindet sich jeweils eine Lücke, die dem DC Zeit zum Umschalten seiner Modi (Lesen und Schreiben) läßt und außerdem für eine symmetrische Verteilung der Sektoren auf Diskette sorgt.

So, und jetzt genug der Wiederholung. Wir werden uns auf ein paar grundsätzliche Programmbeispiele stürzen, die Sie später in eigene Anwendungen einbauen können.

Wie wir schon wissen, werden alle Schreib-/Lesevorgänge des Diskcontrollers interruptgesteuert vorgenommen. Es ist also zum direkten Eingriff auf Diskette notwendig, daß wir uns die Regeln der Interruptsteuerung genau einprägen, da uns die Floppystation bei unseren Experimenten sonst mit Sicherheit »abstürzt«.

Da wir in unserem Kurs verständlicherweise kein DOS-Listing abdrucken können, wurden die wichtigsten Adressen, die wir benötigen, in Tabelle 14 zusammengefaßt und mit einer kurzen Erläuterung versehen, damit Sie sich mit der Anwendung der DOS-Routinen vertraut machen können.

Ein weiteres »Werkzeug« ist die RAM-Belegung der wichtigsten Speicherstellen.

Um den Einstieg zu finden, fangen wir gleich einmal mit der Übergabe der Kommandos an den DC an. Wie bewerkstelligt es das Hauptprogramm, die unterschiedlichsten Befehle wie Lesen, Schreiben, Suchen, Kopf bewegen, Laufwerksmotor an, Formatieren und so weiter an den Diskcontroller zu übergeben?

Um eine Antwort auf diese Frage zu finden, betrachten Sie sich bitte die Tabelle 15. Sie enthält eine Aufstellung aller Jobcodes der Floppy 1541. Mit Jobcodes sind hierbei die Kommandos gemeint, die dafür sorgen, daß ein bestimmter Job zur Ausführung kommt.

Nehmen Sie jetzt einmal die Belegung der Zero-Page zur Hand. Wenn Sie sich die Speicherstellen $0000 bis $0005 betrachten, so merken Sie schon am Namen, daß diese Adressen etwas mit unserer Sache zu tun haben. Es handelt sich hierbei um die Jobspeicher, die die Aufgabe haben, für den Dialog zwischen Hauptprogramm und DC zu sorgen.

Wurde eben etwas von Dialog (nicht etwa Monolog) gesagt? Genau! Die Jobspeicher dienen nicht nur der Übergabe der Kommandos vom Hauptprogramm an den Diskcontroller; sie enthalten nach der Ausführung des Jobs auch die Rückmeldung des DC, an der das Hauptprogramm erkennen kann, ob der Job erfolgreich, das heißt fehlerlos durchgeführt worden ist.

Die Rückmeldungen des Diskcontrollers sind komplett in Tabelle 16 aufgeführt. Wenn Sie sich einmal die Bitmuster der Jobcodes und der Rückmeldungen ansehen und beide Typen miteinander vergleichen, so werden Sie sehr schnell einen Unterschied feststellen, der von entscheidender Bedeutung ist:

Die Befehls-Codes sind ausschließlich negative Werte, das heißt Werte, die größer als $80 (128) sind. Das Kennzeichen solcher Zahlen ist das gesetzte Bit 7 im Byte, das deshalb auch als »negative bit« bezeichnet wird und bei jeder Befehlsausführung in Maschinensprache direkt in das Prozessorstatusregister übernommen wird (N-Flag).

Die Rückmeldungen sind fast ausschließlich Zahlen, die

kleiner als $0F (16) sind (bis auf eine Ausnahme). Diese Größe spielt zwar nicht direkt eine Rolle; das Wichtigste ist jedoch, daß bei diesen Werten keiner größer als $7F (127) ist. Zu der Begründung für diese Einteilung werden wir im folgenden noch kommen.

Wie Sie aus der Belegung der Zero-Page ersehen, existiert für jeden Puffer der Floppystation ein eigener Jobspeicher. Das ermöglicht einen sehr dynamischen Einsatz der Floppystation, der es zum Beispiel erlaubt, mit mehreren Puffern gleichzeitig zu arbeiten.

Eine wichtige Regel sollten Sie sich gleich einprägen, damit später keine Pannen passieren: Wenn Sie einen Jobcode an den DC übergeben, sollten Sie darauf achten, daß der DC für die Ausführung dieses Jobs meistens einen Puffer benötigt. Den Puffer, der dabei zum Beispiel beschrieben wird, wählen Sie durch die Übergabe des Jobcodes in der entsprechenden Speicherstelle aus.

Achtung: Verwenden Sie dabei niemals den Puffer, in dem Sie Ihr Programm abgelegt haben, da dieses sonst unter Umständen gelöscht wird und sich die Diskettenstation auf »mysteriöse« Weise verabschiedet.

Haben Sie also beispielsweise ein Programm ab $0300 (Puffer 0) abgelegt, so sollten Sie sich davor hüten, die Zero-Page-Adresse $0000 als Jobspeicher zu benutzen. Auch als Zwischenspeicher sind die Adressen $0000 bis $0005 nicht unbedingt zu empfehlen, da es sonst zu einer kleinen Katastrophe kommen kann.

Jobcodes des DOS:	
$80	– Lesen eines Blocks in einen Puffer
$90	– Schreiben eines Blocks aus einem Puffer
$A0	– Verify eines Sektors mit einem Pufferinhalt
$B0	– Testen eines Sektors auf Vorhandensein
$C0	– 'Bump' des Tonkopfes
$D0	– Maschinenprogramm im Puffer ausführen
$E0	– Programm im Puffer ausführen, nachdem das Laufwerk hochgefahren ist

Tabelle 15. Zeigt alle Jobcodes mit deren Aufgaben

Rückmeldungen der Jobschleife:	
$01	– Fehlerfreie Durchführung (00, OK)
$02	– Blockheader wurde nicht gefunden (20, Read Error)
$03	– SYNC-Markierung nicht gefunden (21, Read Error)
$04	– Datenblock wurde nicht gefunden (22, Read Error)
$05	– Datenprüfsumme ist falsch (23, Read Error)
$07	– Fehler nach einem Verify (25, Write Error)
$08	– Diskette ist schreibgeschützt (26, Write Peotect on)
$09	– Prüfsumme im Header falsch (27, Read Error)
$0A	– Datenblock auf Diskette zu lang (28, Write Error)
$0B	– Falsche ID im Blockheader (29, Disk ID Mismatch)
$0F	– Keine Diskette im Laufwerk (74, Drive not ready)
$10	– Fehler bei Dekodierung (24, Read Error)

Tabelle 16. Zeigt alle Rückmeldungen des DC, wobei in Klammern die zugehörige Fehlermeldung steht

Die Kommandos an den Diskcontroller

Haben Sie sich die Speicherbelegung der Floppystation schon etwas genauer betrachtet, so werden Ihnen auch die Speicherstellen $0006 bis $0011 nicht entgangen sein.

Wie wir wissen, gibt es verschiedene Jobcodes, die bestimmte Aktionen hervorrufen (die ausführliche Erläuterung der Jobcodes folgt gleich). Nun ist es aber in der Regel notwendig, einem Befehl auch ein paar Parameter mitzugeben, die dann in entsprechender Weise abgearbeitet werden.

In unserem Fall sind das sicherlich die Track- und Sektornummern, auf die sich unser jeweiliger Befehl beziehen soll. Wie Sie aus der Tabelle 15 nämlich ersehen können, existiert zum Beispiel ein Jobcode, der das Lesen eines Blocks veranlaßt. Hier ist es natürlich nötig, die Blockparameter mit anzugeben.

Wollen Sie also ein Kommando $80 an den DC für Puffer 1 übergeben, so schreiben Sie zunächst in die Speicherstelle $0008 die Track-Nummer und in Speicherstelle $0009 die Sektornummer des Blocks, der in Puffer 1 gelesen werden soll. Anschließend erhält die Speicherstelle $0001 den Jobcode und auf geht's... Das klingt alles recht einfach. Stimmt, recht viel komplizierter wird es auch nicht mehr.

Unser einziges Problem besteht jetzt nur noch in der Tatsache, daß der DC für die Ausführung der Befehle eine gewisse Zeit benötigt, die je nach Kommando mehrere Interruptaufrufe erforderlich macht. Woher wissen wir also jetzt, wann ein Block vollständig in den Puffer gelesen ist und wir dessen Inhalt übernehmen können?

Die Lösung dieses Problems liegt in der unterschiedlichen Wertigkeit der Befehls-Bytes und der Rückmeldungen des DC, die ich vorhin schon angesprochen habe. Sie können sich noch erinnern: Alle Jobcodes bestehen aus Werten größer als $80 und alle Rückmeldungen aus Werten kleiner als $80.

Da der DC aber nach jedem Job seine Rückmeldung in der gleichen Speicherstelle hinterläßt, in die wir vorher das Kommando geschrieben hatten, ist es uns nun ein leichtes, diese Speicherstelle zu überprüfen und das Ende des Jobs anhand der Rückmeldung abzufragen. Anhand der noch folgenden Beispiele wird diese Technik gründlich erläutert.

Jetzt wollen wir uns aber mit den eigentlichen Jobcodes und deren Aufgaben beschäftigen.

1) Lesen eines Sektors in einen Puffer:

Wenn wir einen Sektor in einen Puffer lesen wollen, so stellen wir fest, daß diese Aktion auf der Ebene der Jobschleife fast genauso einfach ist, wie von Basic aus mit dem »B-R« beziehungsweise »U1«-Befehl. Zum Lesen eines Sektors geben Sie dessen Track- und Sektornummer in den entsprechenden Speicherstellen für den gewünschten Puffer an. Anschließend senden Sie den Code $80 an den DC, und das Laufwerk startet sofort und liest den Sektor ein.

Diese Befehlsübergabe können Sie sogar von Basic aus, mit den MEMORY- und BLOCK-Befehlen, realisieren und dann den Pufferinhalt auslesen, um sich zu überzeugen, daß der Block auch wirklich eingelesen wurde.

Achtung: Die Diskette muß beim Arbeiten in der Jobschleife von Hand initialisiert werden, da wir uns auf dieser unteren Programmierebene im Rücken der automatischen Initialisierung befinden, die hier deshalb nicht mehr von alleine erfolgt. Merken Sie, daß der Inhalt im Puffer nicht mit dem auf der Diskette übereinstimmt, so kann das mit großer Wahrscheinlichkeit an der fehlenden Initialisierung liegen; doch auch dazu später noch mehr.

Jetzt wollen wir die Jobcodes anhand kleiner Beispiele genauer kennenlernen; dabei wollen wir uns auch gleichzeitig mit den Rückmeldungen des DC vertraut machen, anhand derer sich Fehler in der Ausführung des Jobs erkennen lassen.

Wir werden jetzt den Jobcode für Lesen des Blocks 18,1 in Puffer 0 übergeben und uns dann die Rückmeldung und den Inhalt des Blocks ansehen. Mit dem POKE-Befehl im Programm schreiben wir den Inhalt des Puffers direkt in den Bildschirmspeicher, was für unsere Kontrolle langen soll:

```
1 OPEN 1, 8, 15, "I"
2 PRINT#1, "M-W" CHR$(6) CHR$(0) CHR$(2)
  CHR$(18) CHR$(1)
3 PRINT#1, "M-W" CHR$(0) CHR$(0) CHR$(1)
  CHR$(128)
```

```
4 FORX=0TO2000: NEXT X
5 PRINT#1, "M-R" CHR$(0) CHR$(0) CHR$(1)
6 GET#1, A$: PRINT ASC(A$+CHR$(0))
7 FORX=0TO255
8 PRINT#1, "M-R" CHR$(X) CHR$(3) CHR$(1)
9 GET#1, A$: POKE1024+X,ASC(A$+CHR$(0))
10 NEXT X
11 CLOSE 1
```

Dieses kleine Programm initialisiert die Diskette im Laufwerk. Anschließend werden Track und Sektor (18,1) übergeben und schließlich der Jobcode in Adresse $0000 geschrieben, der dafür sorgt, daß unser Block in Puffer 0 geladen wird. Nach einer kleinen Warteschleife, in der das Laufwerk Zeit zur Befehlsausführung hat, wird der Jobspeicher wieder ausgelesen. Anhand von Tabelle 15 können Sie erkennen, daß der Job ordnungsgemäß ausgeführt wurde, wenn Sie als Rückmeldung eine »1« bekommen.

Auf dem Bildschirm erscheint der Inhalt des Puffers, wobei unter anderem auch Teile des Directory der Diskette zum Vorschein kommen sollten.

2) Schreiben eines Blocks auf Diskette:

Analog zum Lesen eines Blocks erfolgt das Schreiben. Hier übergeben Sie die gleichen Parameter; nur muß sich der zu schreibende Block schon im Puffer der Floppystation befinden. Durch die Auswahl des Job-Speichers können Sie jeden x-beliebigen Puffer des Laufwerks (0 bis 4) in jeden Block der Diskette schreiben.

3) Verifizieren eines Blocks von Diskette:

Dieser Vorgang erfolgt in der Floppystation bei einem SAVE normalerweise automatisch. Aus diesem Grund dauert das Speichern eines Programms auch um einiges länger als das Wiedereinladen in den Computer. Mit Hilfe des entsprechenden Jobcodes ($A0) können wir ein Verify aber nach Belieben starten, um den Inhalt in einem Pufferspeicher mit einem Block auf Diskette zu vergleichen.

Entspricht der Inhalt des Puffers nicht dem Inhalt auf Diskette, so erhalten wir als Rückmeldung die Nummer 7. Beim LOAD-Befehl entspräche das einem »VERIFY ERROR«.

Übrigens: Es wurde ja schon auf die Notwendigkeit des Initialisierens hingewiesen. Unterbleibt dieser Vorgang, so können Sie anhand der Tabelle 16 schon erkennen, was für eine Meldung Sie bekommen werden. Richtig! Die Nummer 11 wird auf Ihrem Bildschirm erscheinen.

4) Suchen eines Sektors:

Dieses Kommando dient nicht dem Lesen eines Blocks von Diskette. Hier wird lediglich untersucht, ob sich der von Ihnen angegebene Block überhaupt auf Diskette befindet. Ist das nicht der Fall, so erhalten Sie eine »2« als Antwort.

Ihnen ist vielleicht auch schon ein weiterer Vorteil der Job-Schleife aufgefallen: Es erfolgt keine Kontrolle auf »legale« Angaben mehr; das heißt, wenn Sie an den Diskcontroller das Kommando geben, daß er Block 2 auf Track 38 lesen soll, dann tut er dies auch.

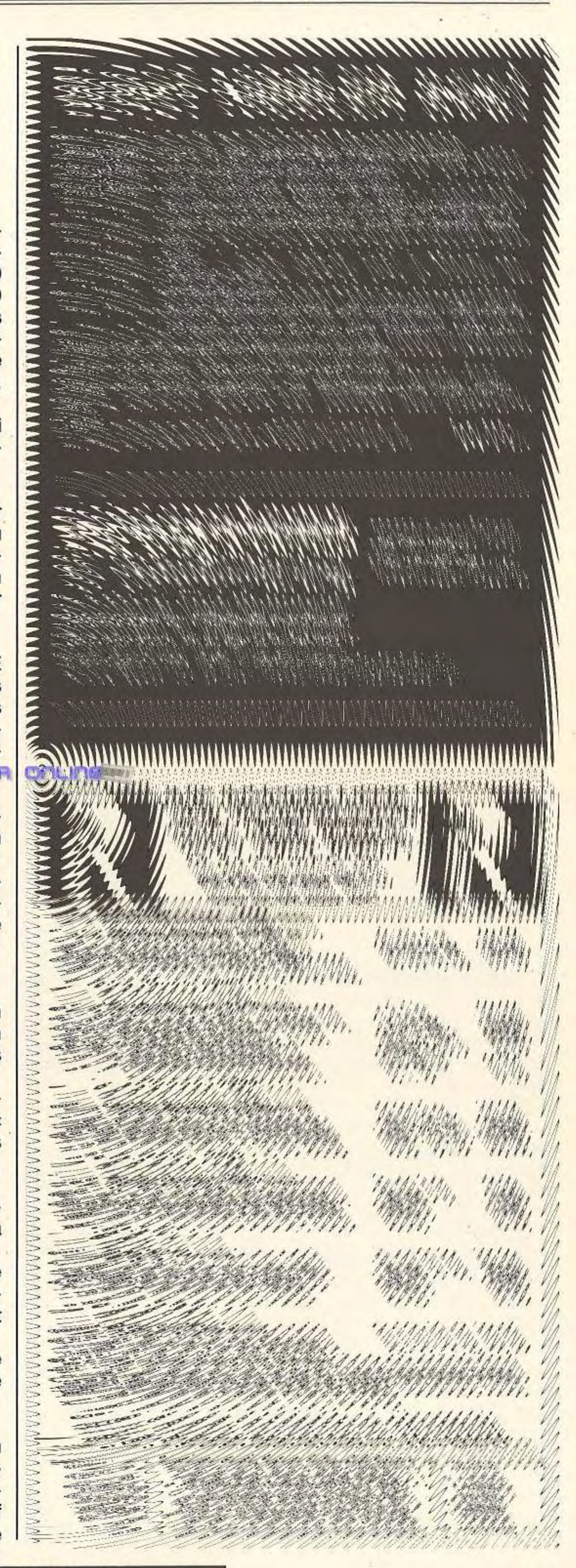
Versuchen Sie das einmal mit dem U1-Befehl; hier bekommen Sie als Antwort: »ILLEGAL TRACK OR SEKTOR«, da Track 38 gar nicht existiert.

So groß der Vorteil dieser Nichtkontrolle auch sein mag; sie sollten sich dessen immer bewußt sein, daß der DC auch versuchen würde, auf Track 100 zuzugreifen, wenn dies verlangt werden sollte.

Die Folge wäre hierbei ein Anschlagen des Kopfes an die vordere Laufschienenbegrenzung der Mechanik; eine sicherlich nicht sehr schonende Angelegenheit.

5. Kopf neu positionieren (Bump):

Dieser Befehl hat eine nützliche Funktion, die jedoch auch für eine sicher nicht unerhebliche Menge an verstellten Tonköpfen verantwortlich ist. Kann der DC einen Track nicht identifizieren, so besteht die Möglichkeit, daß der Kopf sich auf einer illegalen Spur befindet. In diesem Fall kann der DC die



Position des Kopfes nicht mehr anhand der Blockheader auf jedem Track bestimmen.

Aus diesem Grund passiert folgendes: Der DC fährt den Kopf zurück an den Anschlag, und nach einem »Rattern« erfolgt eine neue Ansteuerung des gewünschten Tracks.

Mit dem Kommando $C0 können Sie ein solches Bump ausführen lassen. Nach dem Bump können Sie den Kopf neu positionieren lassen; der Tonkopf steht ansonsten immer auf Track 1.

6) Maschinenprogramm im Puffer starten:

Mit dem Jobcode $D0 machen Sie intern genau das, was extern mit dem M-E-Befehl funktioniert. Der Unterschied zum M-E-Befehl besteht nur in der Tatsache, daß das Programm, das durch $D0 aufgerufen wird, als Interruptprogramm arbeitet, das heißt es wird in die Job-Schleife mit eingebaut und darf deshalb nicht mit einem RTS enden, da ein JMP zurück in die Jobschleife erfolgen muß.

Wie Sie aus einem solchen Programm zurückspringen, wird später noch erläutert.

7) Programm im Puffer starten, nachdem das Laufwerk hochgefahren ist:

Den letzten Befehl werden wir kaum benutzen, da ihm eine Eigenschaft fehlt, die wir dringend benötigen. Wollen wir nämlich ein Programm in der Job-Schleife starten, so werden wir meistens Schreib- oder Lesezugriffe auf die Diskette ausführen. Dies ist jedoch mit $D0 nicht möglich, da das Laufwerk stillsteht.

Der Befehl $E0 hat nun folgende Auswirkungen: Erkennt der DC den Jobcode, so wird das Laufwerk angefahren und die Hardware auf Diskettenzugriff vorbereitet. Mit Hilfe dieses Befehls ist es also möglich, direkt auf die Diskette zuzugreifen, was in einem eigenen Maschinenprogramm erfolgt.

Auch hier muß das Programm mit einem JMP-Befehl beendet werden, da ein Rücksprung in die Job-Schleife erfolgen soll.

Wichtig ist noch, daß das Programm, das mit $D0 oder $E0 gestartet werden soll, immer am Anfang des entsprechenden Puffers stehen muß. Sollen also Programmteile aufgerufen werden, die an höheren Adressen, als $xx00 (xx=03 bis 07) stehen, so müssen diese über Sprungbefehle aufgerufen werden.

Wie schreibt das DOS auf Diskette?

Mit $E0 werden wir uns in unserem Kurs noch öfters beschäftigen, da er die Grundlage der Diskettenzugriffe darstellt (er wird auch vom DOS für das Formatieren angewendet).

Eine Sache dürfen Sie aber auch beim Jobcode $E0 nicht vergessen, nämlich Track- und Sektornummer anzugeben. Es wird, wie schon erwähnt, das Laufwerk betriebsbereit gemacht. Dazu gehört aber auch das Positionieren des Tonkopfes auf die richtige Spur.

Wir haben jetzt die Möglichkeit, ein Maschinenprogramm im Pufferspeicher der Floppy abzulegen und dort zu starten. Unsere Jobcodes erlauben es uns außerdem, direkt in den Ablauf der Job-Schleife einzugreifen und die Diskette sozusagen »von Hand« zu manipulieren.

Als letztes fehlen uns jetzt nur noch die Kenntnisse über den direkten Zugriff auf den Schreib-/Lesekopf der Floppy, so daß wir einzelne Bits ohne Umwege und ohne irgendeine Einschränkung durch die Blockstruktur der Diskette direkt auf die Magnetschicht schreiben können. Mit diesem Problem, das eigentlich gar keines ist, wollen wir uns jetzt beschäftigen. Dazu ein paar Bemerkungen zur Organisation der Schreib-/Leseelektronik der Floppy 1541.

Die Bytes werden zwar auf Diskette in serieller Bitfolge abgelegt; dieses Problem braucht uns jedoch gar nicht weiter zu beschäftigen. Der VIA 6522, der für uns die Elektronik steuert, kann nämlich von uns wie eine Speicherstelle behandelt werden. Senden wir also ein Byte an den VIA 6522, so geschieht das Schreiben auf Diskette vollautomatisch, so daß uns diese Sache nicht weiter beschäftigen soll.

Das einzige Problem, das sich bei der ganzen Angelegenheit stellt, ist die Frage des Timing. Immerhin benötigt der Schreib- oder Lesevorgang eine gewisse Zeit, das heißt, wenn wir beispielsweise Daten vom Tonkopf lesen wollen, muß uns der DC erst mitteilen, wann das nächste Byte fertig eingelesen ist und zur Ausgabe bereitsteht.

Zur Steuerung dieses Timings wird in der Floppy 1541 das V-(Overflow-)Flag des Prozessorstatusregisters benutzt. Der Mikroprozessor 6502 hat nämlich den Vorteil, daß dieses Flag extern beeinflußt werden kann.

Die Regel sieht also folgendermaßen aus: Hat die Lese-Elektronik ein Byte vollständig eingelesen, so wird das V-Flag auf »1« gesetzt. Genauso verhält es sich mit dem Schreiben: Wurde das gegebene Byte komplett auf Diskette geschrieben, so erfolgt ebenfalls ein Setzen des V-Flags.

Das einzige, das der Programmierer nie vergessen darf, ist, daß das V-Flag nach dem Erkennen »von Hand« wieder auf »0« gesetzt werden muß, damit keine Fehlinformation erfolgen kann.

Die Speicherstelle, die für Schreib- und Lesebetrieb zuständig ist, ist Port A des Diskcontrollers mit der Adresse $1C01.

Endlich kommt die Praxis

So, nachdem wir nun so ziemlich alle Voraussetzungen zum Programmieren haben, soll es jetzt endlich mit der praktischen Anwendung unseres Wissens losgehen. Das Werkzeug, das wir jetzt benötigen, besteht aus einem komfortablen Monitor mit »Miniassembler«. Da die Floppy-Programme, die zum Beispiel Fehler auf Diskette bringen, relativ kurz sind, ist es am besten, wenn wir einen Monitor in den Bereich ab $C000 laden und uns anschließend den Bereich ab $8000 für unsere Anwendungen sichern:

POKE 56,127: POKE 52,127: NEW (oder CLR)

Wir legen also unsere kleinen Maschinenprogramme ab $8000 ab und senden diese jeweils mit einem Basic-Programm zur 1541, wo wir sie dann ausführen.

Achtung: Bei einem Reset wird der Speicher der Floppystation gelöscht. Es ist also empfehlenswert, die Programme vor jedem Neustart wieder in den Pufferspeicher des 1541-Laufwerks zu schreiben.

Error Nummer 21 auf Diskette

Ein früher beliebter Programmschutz war das Aufbringen von Errors auf Diskette. Diese konnten von den »alten« Kopierprogrammen nicht übernommen werden. Das geschützte Programm brauchte also nur einen definierten Fehler auf Diskette abzufragen und bei Nichtvorhandensein »auszusteigen«. Wenn Sie sich die Tabelle der Fehlermeldungen im Commodore-Handbuch zur Floppy 1541 ansehen, werden Sie sehr schnell erkennen, daß es für jeden kleinen Defekt eine eigene Fehlernummer gibt. Betrachten Sie jetzt Tabelle 16 dieser Folge, so können Sie dort ablesen, welche Rückmeldung des DC welche Fehlermeldung an den Computer zur Folge hat.

Wir wollen uns einmal den Fehler mit der Nummer 21 ansehen. Er tritt dann auf, wenn die Floppystation versucht, einen Track zu lesen, auf diesem jedoch keine SYNC-Markierungen findet. Das ist zum Beispiel bei einer unformatierten oder beschädigten Diskette der Fall.

Unser kleines Programm in Listing 17 werden Sie vom Prinzip sehr schnell durchschauen. Es macht nichts weiter, als einen bestimmten Track auf Diskette mit lauter $55 (binär: 01010101) zu überschreiben. Das hat zur Folge, daß alle SYNC-Markierungen gelöscht werden und ein Fehler »21« ist die Folge, wenn ein Zugriff stattfinden soll.

Für unsere Versuche sollten Sie eine leere, neuformatierte Diskette verwenden, die Sie sich speziell für unsere Experimente aufheben. Geben Sie also einmal das Programm in Listing 17 ein und starten Sie es anschließend (leere Diskette einlegen!).

Versuchen Sie nun, den Track 1 Ihrer Diskette später einmal zu lesen, so wird sich die Floppy mit einem »21, READ ERROR« dafür bedanken.

Wie Sie sehen, ist ein Fehler 21 recht einfach zu erzeugen, da sich dieser über einen gesamten Track erstreckt (alle Informationen werden gelöscht).

Schwieriger wird es bei anderen Fehlern, die beispielsweise nur in einzelnen Blöcken vorkommen, wobei einige davon (20, 22) auch auf einen gesamten Track geschrieben werden können. Es sind dies die Fehler mit den Nummern 23, 24, 27, 28 und 29.

```
0500        ; LOESCHEN VON TRACK 1
            ; PROGRAMMSTART BEI $0506

0500        JSR $FEOE   ; TRACK LOESCHEN
0503        JMP $FD9E   ; ZUR JOBSCHLEIFE
0506        LDA #$01    ; TRACKNUMMER
0508        STA $0A     ; IN JOBSPEICHER
050A        LDA #$E0    ; JOBCODE
050C        STA $02     ; UEBERGEBEN
050E WAIT   LDA $02     ; RUECKMELDUNG
0510        BMI WAIT    ; ENDE ABWARTEN
0512        RTS         ; PROGRAMMENDE
```

Listing 17. Herstellen eines »21, READ ERROR« auf einer Spur. Startadresse bei $ 0506.

```
  AUF SCHREIBEN UMSCHALTEN

LDA $1C0C    ; PCR
AND #$1F     ; AUF SCHREIBMODUS
ORA #$D0     ;
STA $1C00    ; UMSCHALTEN
LDA #$FF     ;
STA $1C03    ; PORT A AUF AUSGANG

  AUF LESEN UMSCHALTEN (AUCH JSR $FE00)

LDA $1C0C    ; PCR
ORA #$E0     ; AUF LESEMODUS
STA $1C0C    ; UMSCHALTEN
LDA #$00     ;
STA $1C03    ; PORT A AUF EINGANG
```

Listing 18a. Ein READ ERROR 22 wird erzeugt (in einem beliebigen Sektor)

Um solche Fehler zu erzeugen, muß jeweils der zu zerstörende Sektor abgetastet werden, bis die richtige Stelle für den Eingriff gefunden wird. Damit Sie die wichtigen Routinen zur Arbeit innerhalb der Job-Schleife ebenfalls aufrufen können, sind in Tabelle 14 ein paar wichtige Unterprogramme des DOS mit den geforderten Parametern aufgeführt.

Einen Error 22 beispielsweise würden Sie dadurch herstellen, daß Sie die Routine zum Finden des Datenblocks aufrufen. Diese kehrt bei gefundenem Datenblock mit RTS zurück. Jetzt schalten Sie auf Schreiben um (in Bild 6 dargestellt) und bringen ein paar Byte ohne Konzept auf die Diskette. Versucht der DC, diesen Datenblock später einmal zu lesen, so erfolgt ein Fehler 22, da Sie die Datenblockkennung, die direkt hinter der SYNC-Markierung steht, zerstört haben.

```
5 REM PROGRAMM ZUM ERZEUGEN EINES               <209>
6 REM      22, READ ERROR                       <197>
7 :                                             <239>
10 POKE 56,31:POKE 52,31:CLR:OPEN 1,8,15,"
   I"                                           <192>
20 FOR X=0 TO 80:READ A:POKE 32768+X,A:NEX
   T                                            <105>
30 INPUT"{CLR,DOWN,SPACE}TRACK FUER ERROR
   22";T                                        <222>
40 INPUT"{DOWN}SEKTOR FUER ERROR 22";S          <006>
50 POKE 32777,T:POKE 32834,T:POKE 32781,S       <202>
60 RESTORE                                      <110>
70 FOR X=0 TO 80:PRINT#1,"M-W"CHR$(X)CHR$(
   5)CHR$(1)CHR$(PEEK(X+32768)):NEXT            <032>
80 PRINT:PRINT:PRINT"PROGRAMM STARTET"          <052>
90 PRINT#1,"M-E"CHR$(64)CHR$(5):CLOSE 1:EN
   D                                            <056>
100 DATA 165,18,133,22,165,19,133,23,169,3
    5,133,24,169,1,133,25,32,39                 <211>
110 DATA 245,32,86,245,173,12,28,41,31,9,1
    92,141,12,28,169,255,141,3,28               <219>
120 DATA 169,85,141,1,28,80,254,184,80          <108>
130 DATA 254,184,80,254,184,32,0,254,76         <156>
140 DATA 158,253,234,234,234,234,234,234        <103>
150 DATA 234,234,169,35,133,10,169,224,133
    ,2,165,2,48,252,96,0,0,0                    <224>
```

Listing 18b. Herstellen eines »22, READ ERROR« (Assemblerprogramm)

```
5 REM PROGRAMM ZUM ERZEUGEN EINES               <209>
6 REM      23, READ ERROR                       <213>
7 :                                             <239>
10 POKE 56,31:POKE 52,31:CLR:OPEN 1,8,15,"
   I"                                           <192>
20 FOR X=0 TO 80:READ A:POKE 32768+X,A:NEX
   T                                            <105>
30 INPUT"{CLR,DOWN,SPACE}TRACK FUER ERROR
   23";T                                        <254>
40 INPUT"{DOWN}SEKTOR FUER ERROR 23";S          <038>
50 POKE 32777,T:POKE 32834,T:POKE 32781,S       <202>
60 RESTORE                                      <110>
70 FOR X=0 TO 80:PRINT#1,"M-W"CHR$(X)CHR$(
   5)CHR$(1)CHR$(PEEK(X+32768)):NEXT            <032>
80 PRINT:PRINT:PRINT"PROGRAMM STARTET"          <052>
90 PRINT#1,"M-E"CHR$(64)CHR$(5):CLOSE 1:EN
   D                                            <056>
100 DATA 165,18,133,22,165,19,133,23,169,3
    5,133,24,169,0,133,25,32,39                 <209>
110 DATA 245,32,86,245,162,0,202,208,253        <229>
120 DATA 173,12,28,41,31,9,192,141,12,28,1
    69,255,141,3,28,169,85,141,1                <121>
130 DATA 28,80,254,184,80,254,184,80,254,1
    84,32,0,254,76,158,253,234,234              <084>
140 DATA 234,169,35,133,10,169,224,133,2,1
    65,2,48,252,96,0,0,0                        <083>
```

Listing 19a. Ein READ ERROR 23 wird erzeugt

Wollen Sie einen Fehler mit der Nummer 23, dann ist es erforderlich, daß Sie den Vorspann des Datenblocks überspringen und erst inmitten der gespeicherten Daten einen Schreibzugriff durchführen. Durch diesen Zugriff, der in der Prüfsumme am Blockende natürlich nicht verzeichnet wird, folgt die Meldung »23, READ ERROR«, als Zeichen eines Prüfsummenfehlers.

Listing 18a und 19a zeigen Ihnen Programme, die einen Error 22 und einen Error 23 erzeugen (Listing 18b und 19b sind die zugehörigen Quellprogramme).

Der Vorteil eines Fehlers mit der Nummer 23, ist daß die Daten in der Regel schon im Puffer stehen, bevor der Fehler erkannt wird, das heißt, Sie können einen Datenblock auf Diskette gezielt mit einem Fehler versehen, obwohl dieser noch lesbare Inhalte enthält.

Die eben besprochenen Fehler auf Diskette eignen sich hervorragend für einen Kopierschutz. Am wirkungsvollsten

sind dabei mit Sicherheit solche Fehler, die zusätzlich noch Daten enthalten. Es gibt nämlich schon eine ganze Menge von Programmen, die Fehler übernehmen und auf der Kopie wieder simulieren.

Soweit zu Fehlern. Haben Sie schon einmal etwas von »Killertracks« gehört? Dieses anschauliche Wort steht für die Manipulation eines Tracks, der sämtliche Sicherheitseinrichtungen des DOS durcheinanderbringt.

Vielleicht hatten Sie schon einmal eine Diskette in Ihren Händen, die folgendes »Phänomen« aufzeigte: Wenn Sie versuchten, einen Block auf einer bestimmten Spur zu lesen, ist der Schreib-/Lesekopf der Floppystation ordnungsgemäß auf den Track positioniert worden. Danach hat der DC mit dem Lesen des Blocks angefangen und – nicht mehr aufgehört. Anders ausgedrückt: Die Floppy 1541 las und las ...

Die Spur, die Sie da versucht haben zu lesen, hat grundsätzlich dafür gesorgt, daß sich die Diskettenstation »aufhängte«. Daß es sich hier um den schon angesprochenen »Killertrack« handelte, brauche ich kaum noch zu erwähnen. Aber, wie stellt man eine solche »Falle« her? Was ist mit dem Track passiert, daß der DC völlig »aus dem Häuschen« gerät? Die Antwort sehen Sie in Listing 20. Dieses kleine Programm stellt einen solchen »Killertrack« her. Des Rätsels Lösung ist eigentlich ganz einfach: Die gesamte Spur besteht aus einer einzigen SYNC-Markierung. Da die SYNC-Markierung von der Lese-/Schreibelektronik speziell verarbeitet wird, verzögert sich die Arbeit des DC gewaltig, wenn eine solche »Dauer-SYNC-Markierung« auftritt.

Da die Floppystation bei Fehlern bis zu über 200mal versucht, einen Block zu lesen, dehnt sich der Zeitraum, den sie bei Verzögerungen benötigt, stark aus. Bei Killertracks braucht die Diskettenstation pro Leseversuch eine Unmenge an Zeit, was sich auch im langsamen Blinkrhythmus der LED am Laufwerk zeigt.

Allein schon an den kleinen Anwendungen können Sie erkennen, wie vielseitig und vielfältig die Möglichkeiten sind, die einem in der Programmierung offenstehen. Wenn Sie intensiv mit der Floppystation arbeiten, werden Sie bald schon neue Anwendungsmöglichkeiten kennenlernen. Aus der Floppy 1541 läßt sich noch eine Menge herausholen, wie wir noch feststellen werden, wobei der Kopierschutz von Disketten sicher nur einen kleinen Teil der vielfältigen Möglichkeiten darstellt.

```
0500      LDA $12     ; ID 1 HOLEN
0502      STA $16     ; UND UEBERNEHMEN
0504      LDA $13     ; ID 2 HOLEN
0506      STA $17     ; UND UEBERNEHMEN
0508      LDA #$23    ; TRACKNUMMER
050A      STA $18     ; UEBERNEHMEN
050C      LDA 01      ; SEKTORNUMMER
050E      STA $19     ; UEBERNEHMEN
0510      JSR $F527   ; BLOCKHEADER HOLE
                        N
0513      JSR $F556   ; AUF 'SYNC' WARTE
                        N
0516      LDA $1C0C   ; PCR
0519      AND #$1F    ; AUF SCHREIBEN
051B      ORA #$C0    ; UMSCHALTEN
051D      STA $1C0C   ;
0520      LDA #$FF    ; PORT A AUF AUSGA
                        NG
0522      STA $1C03   ; UMSCHALTEN
0525      LDA #$55    ; FALSCHWERT
0527      STA $1C01   ;
052A W1   BVC W1      ; SCHREIBEN
052C      CLV         ;
052D W2   BVC W2      ; SCHREIBEN
052F      CLV         ;
0530 W3   BVC W3      ; SCHREIBEN
0532      CLV         ;
0533      JSR $FE00   ; PCR AUF LESEN
0536      JMP $FD9E   ; ZUR JOBSCHLEIFE
0539      NOP         ;
053A      NOP         ;
053B      NOP         ;
053C      NOP         ;
053D      NOP         ;
053E      NOP         ;
053F      NOP         ;
0540      NOP         ;
0541      LDA #$23    ; TRACKNUMMER
0543      STA $0A     ; IN JOBSPEICHER
0545      LDA #$E0    ; JOBCODE
0547      STA $02     ; UEBERGEBEN
0549 WAIT LDA $02     ; RUECKMELDUNG
054B      BMI WAIT    ; WARTEN AUF ENDE
054D      RTS         ; PROGRAMMENDE
```

Listing 18b. Herstellen eines »22, READ ERROR« (Assemblerprogramm)

Formatieren – was ist das?

Wie jedem Floppy-Besitzer bekannt ist, muß eine Diskette vor dem ersten Speichern von Daten formatiert werden. Wie eine Diskette nach einem solchen Formatiervorgang aussieht, wurde schon besprochen.

Uns soll nun interessieren, was während des Formatierens so alles in der Floppystation passiert, und warum die 1541 so lange für einen eigentlich sehr einfachen Vorgang benötigt.

Zur Wiederholung: Beim Formatieren werden vom DOS alle wichtigen Markierungen auf die Diskette gebracht und außerdem sämtliche Sektoren in ihrer späteren Form angelegt.

Der Vorgang des Formatierens verwendet zu seiner Ausführung einen uns schon bekannten Jobcode, nämlich $E0.

Bevor das DOS den eigentlichen Formatiervorgang startet, wird ab $0600 (also im Puffer 3) ein Sprungbefehl abgelegt: JMP $FAC7.

Dieser Sprungbefehl ist eine Art Vektor, der im RAM liegt und somit verändert werden kann. Er bietet dem Benutzer die Möglichkeit, eine eigene Routine einzubauen, die dann bei jedem Track-Wechsel angesprungen wird, um so einige wirksame Manipulationen an der Formatierung vorzunehmen, indem zum Beispiel Werte in der Zero-Page verändert werden, doch dazu später. Üblicherweise zeigt dieser Vektor direkt auf eine Jobroutine, die für das Formatieren zuständig ist. Diese Routine wird nun vom Hauptprogramm mit dem Jobcode $E0, der in Speicherstelle $03 geschrieben wird, aufgerufen.

Formatieren in der Job-Schleife

Am Anfang der Jobroutine steht nun die Abfrage, ob schon mindestens ein Track formatiert wurde oder ob dieser Einsprung der allererste ist. Ist dieser Einsprung der erste, so werden alle Parameter für den Steppermotor gesetzt; danach erfolgt ein Rücksprung in die Job-Schleife. Hier wird der Tonkopf nun 45 (!) Tracks zurückgefahren, was sich in jenem charakteristischen Rattern der Floppystation äußert.

Nun, können Sie sagen, es würde auch reichen, wenn der Kopf nur 35 oder 40 Spuren zurücktransportiert würde. In der Tat ist der Wert 45 sehr hoch. Man muß aber bedenken, daß es passieren kann, daß der Schreib-/Lesekopf der Floppystation durch eine defekte Diskette oder einen Programmierfehler zu weit nach innen gefahren und beispielsweise auf Track 42 am Anschlag gelandet ist und daß ein Zurückfahren um 40 Tracks einfach nicht ausreicht, um den Tonkopf richtig zu positionieren.

```
0500      LDA $12     ; ID 1 HOLEN
0502      STA $16     ; UND UEBERNEHMEN
0504      LDA $13     ; ID 2 HOLEN
0506      STA $17     ; UND UEBERNEHMEN
0508      LDA #$23    ; TRACKNUMMER
050A      STA $18     ; UEBERNEHMEN
050C      LDA 00      ; SEKTORNUMMER
050E      STA $19     ; UEBERNEHMEN
0510      JSR $F527   ; BLOCKHEADER HOLE
                        N
0513      JSR $F556   ; AUF 'SYNC' WARTE
                        N
0516      LDX #$00    ; WARTEN, UM IN
0518 LOOP DEX         ; DEN DATENBLOCK
0519      BNE LOOP    ; ZU KOMMEN
051B      LDA $1C0C   ;
051E      AND #$1F    ; PCR AUF SCHREIBE
                        N
0520      ORA #$C0    ; UMSCHALTEN
0522      STA $1C0C   ;
0525      LDA #$FF    ; PORT A AUF
0527      STA $1C03   ; AUSGANG STELLEN
052A      LDA #$55    ; FALSCHWERT
052C      STA $1C01   ; IN PUFFER SCHREI
                        BEN
052F W1   BVC W1      ; WARTEN AUF READY
0531      CLV         ; FLAG LOESCHEN
0532 W2   BVC W2      ; WARTEN AUF READY
0534      CLV         ; FLAG LOESCHEN
0535 W3   BVC W3      ; WARTEN AUF READY
0537      CLV         ; FLAG LOESCHEN
0538      JSR $FE00   ; AUF LESEN SCHALT
                        EN
053B      JMP $FD9E   ; ZUR JOBSCHLEIFE
053E      NOP         ;
053F      NOP         ;
0540      NOP         ;
0541      LDA #$23    ; TRACKNUMMER
0543      STA $0A     ; IN JOBSPEICHER
0545      LDA #$E0    ; JOBCODE
0547      STA $02     ; UEBERGEBEN
0549 WAIT LDA $02     ; RUECKMELDUNG
054B      BMI WAIT    ; WARTEN AUF ENDE
054D      RTS         ; PROGRAMMENDE
```

Listing 19b. Herstellen eines »23, READ ERROR« (Assemblerprogramm)

```
0500      JSR $FDA3   ; TRACK LOESCHEN
0503      JMP $FD9E   ; ZUR JOBSCHLEIFE
0506      LDA #$01    ; TRACKNUMMER
0508      STA $0A     ; IN JOBSPEICHER
050A      LDA #$E0    ; JOBCODE
050C WAIT STA $02     ; UEBERGEBEN
050E      LDA $02     ; RUECKMELDUNG
0510      BMI WAIT    ; ENDE ABWARTEN
0512      RTS         ; PROGRAMMENDE
```

Listing 20. Ein »Killertrack« wird erzeugt. Startadresse bei $0506.

Der Wert von 45 Tracks enthält also eine Sicherheitsreserve, die ein Positionieren auf Spur 1 mit Sicherheit ermöglicht.

Wurde der Kopf also auf Track 1 positioniert, so erfolgt erneut ein Einsprung in die Formatierungsroutine; eine Speicherstelle zeigt jetzt an, daß der Kopf auf Track 1 positioniert wurde und das Formatieren starten kann.

Jetzt wird noch geprüft, ob auf die nächste Spur umgeschaltet werden soll, da die aktuelle bereits formatiert wurde (wenn ja, erfolgt wieder ein Einsprung in die Job-Schleife, um das Nötige zu tun).

Diese Abfragen am Anfang der Formatierungsroutine scheinen umständlich und überflüssig zu sein; das Gefühl täuscht jedoch. Wir dürfen ja nicht vergessen, daß die Routine immer nur jeweils einen Track formatiert und danach zur Job-Schleife zurückkehrt, damit der Tonkopf weitergeführt werden kann. Wir haben also gewissermaßen eine Endlosschleife, die nur durch die Feststellung, daß Spur 35 fertig formatiert wurde, beendet wird.

Ausmessen einer Spur

Jetzt haben wir aber endlich alle Voraussetzungen zum Formatieren eines Tracks erfüllt und wollen an die Arbeit gehen. Der Abschnitt, der jetzt besprochen wird, ist übrigens für die langwierige Formatierung verantwortlich und sorgt für die ausgedehnten Wartezeiten.

Bevor die SYNC-Markierungen und Sektoren auf eine Spur geschrieben werden, wird diese Spur vom DOS »ausgemessen«.

Das Betriebssystem der Floppy 1541 »weiß« im Normalfall genau, wieviele Bytes für die SYNC-Markierungen und Sektoren einer Spur benötigt, beziehungsweise verbraucht werden.

»Löcher« zwischen Sektoren

Jetzt ist es aber so, daß die Sektoren nicht genau auf jede Spur abgemessen sind; vielmehr hat die Diskette pro Spur eine etwas höhere Kapazität, als eigentlich benötigt wird. Aus dieser Tatsache folgt natürlich, daß zwischen den einzelnen Sektoren »Leerstellen« entstehen, die keine Daten enthalten.

Da jetzt aber die Länge der Tracks von außen (Track 1) nach innen (Track 35) kontinuierlich abnimmt, werden diese Leerstellen immer kleiner; wir haben also unterschiedliche Anzahlen von »Leer-Bytes« zwischen den Sektoren.

Das DOS ist nun bestrebt, die Sektoren jeder Spur möglichst symmetrisch anzuordnen, also immer den gleichen Abstand zwischen zwei Sektoren eines Tracks zu haben. Bild 7 zeigt, was passiert, wenn keine vorherige Ausmessung stattfindet.

Um das Ziel einer »symmetrisch« formatierten Diskette (Bild 8) zu erreichen, stellt das DOS durch einige komplizierte Schreib- und Lesevorgänge das Verhältnis zwischen benötigtem und vorhandenem Platz einer Spur fest. Aus diesem Verhältnis kann nun anhand einer einfachen Rechnung festgestellt werden, wieviel Platz zwischen den einzelnen Sektoren freigelassen werden muß.

Nachdem diese komplizierte Vermessung stattgefunden hat, die mehrere Diskettenumdrehungen und damit Zeit erfordert, beginnt nun das eigentliche Formatieren der Diskette, das mit allem Drum und Dran normalerweise nicht mehr als eine ⅓ Sekunde für einen Track benötigt.

Das Anlegen der Sektoren im Puffer

Bevor geschrieben werden kann, müssen die Sektoren erst einmal im Pufferspeicher der Floppy 1541 hergestellt werden. Da sich die einzelnen Sektoren nur durch deren Header unterscheiden, reicht das Anlegen der Blockheader; die Inhalte der Datenblöcke sind bei jedem Sektor gleich und bestehen aus dem schon bekannten Muster $4B gefolgt von 255 $01-Byte.

Die Blockheader werden alle in einem Pufferspeicher ($0300-$03FF) abgelegt; der Inhalt der Datenblöcke steht ab $0500 bis $05FF.

Schreiben eines Tracks auf Diskette

So, alle Vorarbeiten wären jetzt abgeschlossen. Wir können mit dem Schreiben auf Diskette beginnen. Zuerst wird

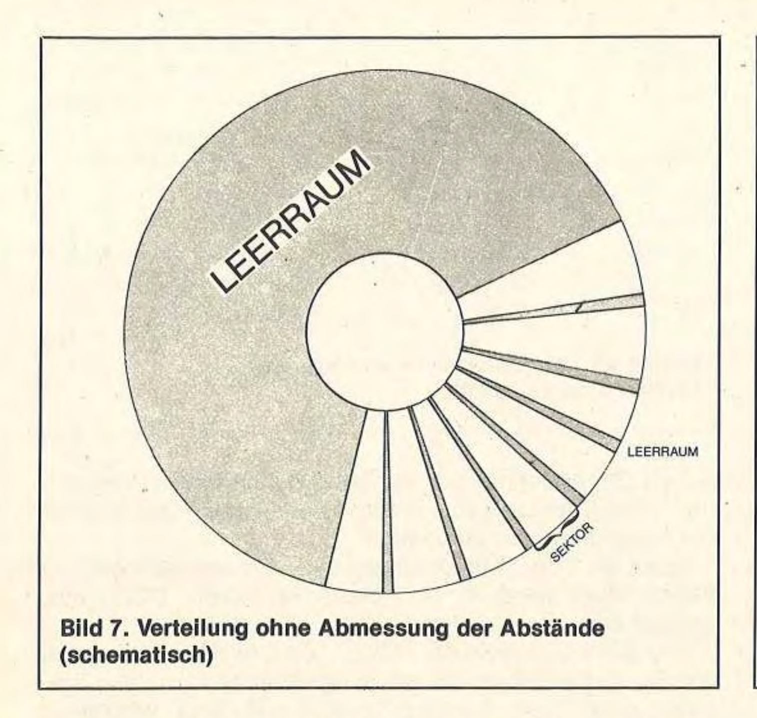


Bild 7. Verteilung ohne Abmessung der Abstände (schematisch)

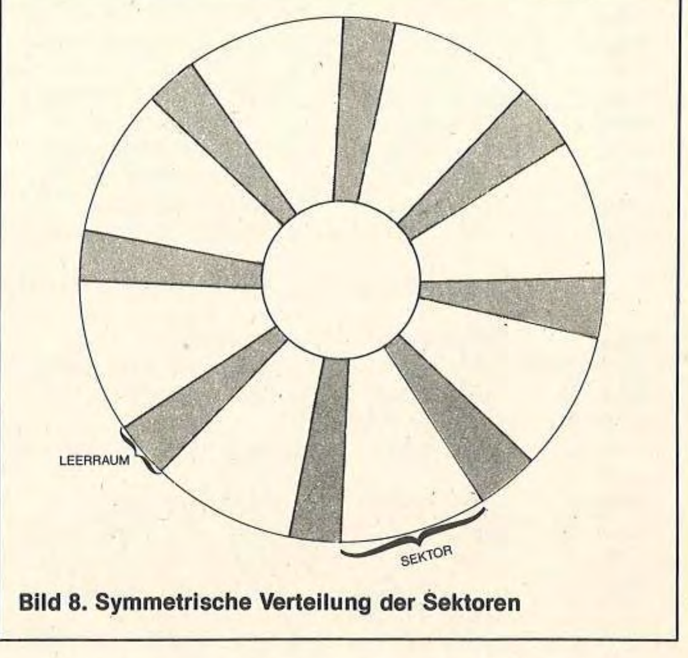


Bild 8. Symmetrische Verteilung der Sektoren

der Diskcontroller auf Schreibmodus gestellt und die Spur der Diskette gelöscht.

Der gesamte Spurinhalt wird nun während einer einzigen Diskettenumdrehung (⅕ Sekunde) auf die Diskette gebracht, wobei zuerst die SYNC-Markierung für den Blockheader, danach der Blockheader selbst, geschrieben werden. Nach einer Lücke von 9 Byte folgt die SYNC-Markierung des Datenblocks mit dem zugehörigen Daten-Byte. Den Abschluß eines Sektors bildet der schon erwähnte »Leerraum«, der aus der vorher errechneten Anzahl von Bytes besteht.

Zur Sicherheit erfolgt nach dem Schreiben eine Verify-Routine, die auf eventuelle Disketten- oder Schreibfehler kontrolliert und bei deren Auftreten einen »24, READ ERROR« ausgibt.

Mit dieser letzten Maßnahme ist eine Spur einer Diskette fertig formatiert worden, und es wird auf Erreichen der Spur 35 abgefragt. Wurde Spur 35 formatiert, so werden alle Flags für das Formatieren zurückgesetzt, die Jobschleife verlassen und ins Hauptprogramm zurückgekehrt.

Im Hauptprogramm wird nun auf Track 18 positioniert. Die BAM der Diskette wird hergestellt und in Block 18,0 abgelegt. Anschließend wird noch der erste Directory-Block (18,1) mit Nullen vollgefüllt und ebenfalls gespeichert, womit das Formatieren abgeschlossen wäre.

Formatieren mit Variationen

Formatiert man eine Diskette nur kurz, das heißt ohne Angabe einer ID beim N-Befehl, so werden alle anfänglichen Schritte weggelassen. Es wird in diesem Fall nur auf das richtige Formatkennzeichen in der BAM ($41/65/A) kontrolliert und danach der eben beschriebene Vorgang auf Track 18 durchgeführt.

Nun wäre unser Floppy-Kurs natürlich kein Floppy-Kurs, wenn wir unsere neu gewonnenen theoretischen Kenntnisse nicht sofort in die Praxis umsetzen wollten.

In der Tat kann man mit Hilfe der Formatierroutine im DOS einige nette »Scherze« auf eine Diskette bringen, die entweder dem Spieltrieb oder dem Softwareschutz dienen können.

Es wurde vorhin schon erwähnt, daß die Formatierroutine jeweils über einen Sprungbefehl bei $0600 im RAM der Floppystation aufgerufen wird.

Diese Adresse wird bei jedem neuen Track angesprungen und bietet so die Möglichkeit, Tracks zu erzeugen, die in ihrem Aufbau voneinander abweichen, wenn entsprechende Eingriffe vorgenommen werden.

Diese Möglichkeit eines Eingriffes wollen wir an dieser Stelle aber gar nicht erst weiter diskutieren, da es ziemlich aussichtslos ist, hier ohne dokumentiertes DOS-Listing an die Arbeit zu gehen.

Daß wir kein DOS-Listing besitzen, soll aber noch lange nicht heißen, daß wir nicht in der Lage sind, auf anderem Weg Eingriffe in die Formatierung vorzunehmen. Wenn wir nicht effektiv mit der fest eingebauten Routine zusammenarbeiten können, dann schreiben wir uns eben ein vollständig eigenes Programm, das im RAM der Floppystation abgelegt wird und uns für Abänderungen unendlich viele Möglichkeiten bietet.

Formatierung »selbst gebaut«

Sehen Sie sich einmal Listing 21 an. Es wurde hier ein Formatiersystem entwickelt, das einfacher und schneller arbeitet als die DOS-Routine und trotzdem ein paar zusätzliche Möglichkeiten bietet.

Da das Gesamtprinzip aber fast 100prozentig mit der im DOS eingebauten Routine übereinstimmt, können Sie sich anhand des Source-Code-Listings einmal die »praktische Ausführung« einer Formatierroutine ansehen.

Um Ihnen die Eingabe des Programms zu erleichtern, wurde ein DATA-Lader als Listing 22 beigefügt, wobei wir Ihnen empfehlen möchten, dieses gleich einmal einzutippen.

Das Programm wird nur aktiviert, wenn alle DATAs richtig eingetippt wurden. Haben Sie alles richtig gemacht, so steht nach der Ausführung des Laders ein Maschinenprogramm am Basic-Anfang, dem eine Basic-Zeile beigefügt ist. Das Programm sollten Sie sich jetzt mit SAVE auf eine Diskette speichern und danach mit RUN starten.

Nach einer winzigen Verzögerung erscheint die READY-Meldung und der Cursor wieder. Das Formatierungsprogramm wurde jetzt in den Bereich ab $C000 (49152) geschoben und der SAVE-Vektor abgeändert.

Tippen Sie jetzt einfach den Befehl SAVE – ohne Anführungszeichen und Filenamen – ein und drücken Sie <RETURN>. Es erscheint nun die Startmeldung des Formatprogrammes. Sie können jetzt einen Namen für eine Dis-

```
Floppyprogramm zum Disk-Format-System
    1985 by KOSS
0500  ea        nop
0501  a5 0a     lda $0a        Tracknummer aus Jobspeicher
0503  c9 24     cmp #$24       größer als 35?
0505  90 07     bcc $050e      nein
0507  a9 12     lda #$12       ja: 18 als Anzahl der Sektoren
0509  85 43     sta $43        festlegen
050b  4c 13 05  jmp $0513
050e  20 4b f2  jsr $f24b      Anzahl der Sektoren holen
0511  85 43     sta $43        und merken
0513  a9 00     lda #$00
0515  85 1b     sta $1b        Sektorzähler setzen
0517  a0 00     ldy #$00
0519  a2 00     ldx #$00
051b  a5 39     lda $39        Kennzeichen $08 für Blockheader
051d  99 00 03  sta $0300,y
0520  c8        iny
0521  c8        iny
0522  a5 1b     lda $1b        Sektornummer
0524  99 00 03  sta $0300,y
0527  c8        iny
0528  a5 0a     lda $0a        Tracknummer
052a  99 00 03  sta $0300,y
052d  c8        iny
052e  a5 13     lda $13        ID 2
0530  99 00 03  sta $0300,y
0533  c8        iny
0534  a5 12     lda $12        ID 1
0536  99 00 03  sta $0300,y
0539  c8        iny
053a  a9 0f     lda #$0f
053c  99 00 03  sta $0300,y    Lücke lassen
053f  c8        iny
0540  99 00 03  sta $0300,y
0543  c8        iny
0544  a9 00     lda #$00
0546  59 fa 02  eor $02fa,y    Prüfsumme bilden
0549  59 fb 02  eor $02fb,y
054c  59 fc 02  eor $02fc,y
054f  59 fd 02  eor $02fd,y
0552  99 f9 02  sta $02f9,y    und abspeichern
0555  e6 1b     inc $1b
0557  a5 1b     lda $1b        Sektorzähler erhöhen
0559  c5 43     cmp $43        schon Maximalzahl?
055b  90 be     bcc $051b      nein, weitermachen
055d  a9 03     lda #$03
055f  85 31     sta $31
0561  98        tya
0562  48        pha
0563  8a        txa
0564  9d 00 07  sta $0700,x    Datenblock mit $00 füllen
0567  e8        inx
0568  d0 fa     bne $0564
056a  20 30 fe  jsr $fe30
056d  68        pla
056e  a8        tay
056f  88        dey
0570  20 e5 fd  jsr $fde5
0573  20 f5 fd  jsr $fdf5
0576  a9 07     lda #$07
0578  85 31     sta $31
057a  20 e9 f5  jsr $f5e9      Prüfsumme für Datenblock
057d  85 3a     sta $3a        abspeichern
057f  20 8f f7  jsr $f78f
0582  a9 00     lda #$00       Sektorzähler setzen
0584  85 32     sta $32
0586  20 0e fe  jsr $fe0e      Track löschen
0589  a9 ff     lda #$ff
058b  8d 01 1c  sta $1c01      SYNC schreiben
058e  a2 05     ldx #$05
0590  50 fe     bvc $0590
0592  b8        clv
0593  ca        dex
0594  d0 fa     bne $0590
0596  a2 0a     ldx #$0a
0598  a4 32     ldy $32
059a  50 fe     bvc $059a      Blockheader schreiben
059c  b8        clv
059d  b9 00 03  lda $0300,y
05a0  8d 01 1c  sta $1c01
05a3  c8        iny
05a4  ca        dex
05a5  d0 f3     bne $059a
05a7  a2 09     ldx #$09
05a9  50 fe     bvc $05a9      Lücke von 9 Bytes lassen
05ab  b8        clv
05ac  a9 55     lda #$55
05ae  8d 01 1c  sta $1c01
05b1  ca        dex
05b2  d0 f5     bne $05a9
05b4  a9 ff     lda #$ff
05b6  a2 05     ldx #$05
05b8  50 fe     bvc $05b8      SYNC-Markierung für Datenblock
05ba  b8        clv
05bb  8d 01 1c  sta $1c01
05be  ca        dex
05bf  d0 f7     bne $05b8
05c1  a2 bb     ldx #$bb
05c3  50 fe     bvc $05c3
05c5  b8        clv
05c6  bd 00 01  lda $0100,x
05c9  8d 01 1c  sta $1c01
05cc  e8        inx
05cd  d0 f4     bne $05c3
05cf  a0 00     ldy #$00
05d1  50 fe     bvc $05d1      Datenblock schreiben
05d3  b8        clv
05d4  b1 30     lda ($30),y
05d6  8d 01 1c  sta $1c01
05d9  c8        iny
05da  d0 f5     bne $05d1
05dc  a9 55     lda #$55
05de  a2 08     ldx #$08       Lücke nach Sektor mit fester
05e0  50 fe     bvc $05e0      Länge von 8 Bytes schreiben.
05e2  b8        clv
05e3  8d 01 1c  sta $1c01
05e6  ca        dex
05e7  d0 f7     bne $05e0
05e9  a5 32     lda $32
05eb  18        clc
05ec  69 0a     adc #$0a
05ee  85 32     sta $32
05f0  c6 1b     dec $1b        schon alle Sektoren?
05f2  d0 95     bne $0589      nein, weitermachen
05f4  50 fe     bvc $05f4
05f6  b8        clv
05f7  50 fe     bvc $05f7
05f9  b8        clv
05fa  20 00 fe  jsr $fe00      auf Lesen umschalten
05fd  a9 c8     lda #$c8       200 Leseversuche
05ff  85 1f     sta $1f
0601  a9 00     lda #$00
0603  85 30     sta $30
0605  a9 03     lda #$03
0607  85 31     sta $31
0609  a5 43     lda $43
060b  85 1b     sta $1b        Sektorzähler
060d  20 56 f5  jsr $f556      auf SYNC-Signal warten
0610  a2 0a     ldx #$0a
0612  a0 00     ldy #$00
0614  50 fe     bvc $0614
0616  b8        clv
0617  ad 01 1c  lda $1c01
061a  d1 30     cmp ($30),y    Daten vergleichen
061c  d0 0e     bne $062c
061e  c8        iny
061f  ca        dex
0620  d0 f2     bne $0614
0622  18        clc
0623  a5 30     lda $30
0625  69 0a     adc #$0a
0627  85 30     sta $30
0629  4c 35 06  jmp $0635
062c  c6 1f     dec $1f        Zähler vermindern
062e  d0 d1     bne $0601
0630  a9 06     lda #$06
0632  4c d3 fd  jmp $fdd3      24, READ ERROR
0635  20 56 f5  jsr $f556      SYNC-Signal abwarten
0638  a0 bb     ldy #$bb
063a  50 fe     bvc $063a
063c  b8        clv
063d  ad 01 1c  lda $1c01
0640  d9 00 01  cmp $0100,y
0643  d0 e7     bne $062c
0645  c8        iny
0646  d0 f2     bne $063a
0648  a2 fc     ldx #$fc
064a  50 fe     bvc $064a
064c  b8        clv
064d  ad 01 1c  lda $1c01
0650  d9 00 07  cmp $0700,y    Datenblock testen
0653  d0 d7     bne $062c
0655  c8        iny
0656  ca        dex
0657  d0 f1     bne $064a
0659  c6 1b     dec $1b
065b  d0 b0     bne $060d
065d  4c 9e fd  jmp $fd9e      Ende; zur Jobschleife
0660  a0 00     ldy #$00       Start des Floppyprogramms
0662  b9 e0 06  lda $06e0,y
0665  99 00 02  sta $0200,y    Disknamen übernehmen
0668  c8        iny
0669  cc df 06  cpy $06df
066c  90 f4     bcc $0662
066e  ad df 06  lda $06df
0671  8d 74 02  sta $0274      Länge der Zeile setzen
0674  ad de 06  lda $06de
0677  8d 7b 02  sta $027b      Kommaposition setzen
067a  a9 00     lda #$00
067c  85 7f     sta $7f        Drive 0 setzen
067e  20 00 c1  jsr $c100      LED am Laufwerk an
0681  ac 7b 02  ldy $027b
0684  b9 00 02  lda $0200,y    ID 1 holen
0687  85 12     sta $12
0689  b9 01 02  lda $0201,y    id 2 holen
068c  85 13     sta $13
068e  20 07 d3  jsr $d307      alle Kanäle schließen
0691  a9 1a     lda #$1a
0693  8d 05 1c  sta $1c05      Timer setzen
0696  a9 c0     lda #$c0       BUMP anfordern
0698  85 00     sta $00
069a  a5 00     lda $00
069c  30 fc     bmi $069a      auf Ausführung warten
069e  ae dc 06  ldx $06dc      erste Tracknummer
06a1  86 0a     stx $0a
06a3  a9 e0     lda #$e0       Track formatieren
06a5  85 02     sta $02
06a7  a5 02     lda $02
06a9  30 fc     bmi $06a7      auf Ende warten
06ab  c9 02     cmp #$02       Fehler aufgetreten?
06ad  b0 0c     bcs $06bb      verzweige, wenn ja
06af  e8        inx
06b0  ec dd 06  cpx $06dd      schon Zieltrack formatiert?
06b3  90 ec     bcc $06a1      weiter, wenn nein
06b5  20 40 ee  jsr $ee40      Directory herstellen
06b8  60        rts            Ende
```

Listing 21. Eine neue Formatierroutine.

```
06b9  ea        nop
06ba  ea        nop
06bb  a2 02     ldx #$02
06bd  4c 0a e6  jmp $e60a      Diskstatus ausgeben; Ende

  Computerprogramm zum Disk-Format-System
  (c) 1985 by KOSS
., c200 a2 00     ldx #$00
., c202 20 87 c2  jsr $c287     Titel und erste Frage ausgeben
., c205 a0 00     ldy #$00
., c207 20 cf ff  jsr $ffcf     Eingabe holen
., c20a c9 0d     cmp #$0d
., c20c f0 08     beq $c216
., c20e 99 e0 c1  sta $c1e0,y   Namen abspeichern
., c211 c8        iny
., c212 c0 10     cpy #$10      schon 16 Zeichen ?
., c214 90 f1     bcc $c207     weiter, wenn nein
., c216 a9 2c     lda #$2c      Komma hinter den Namen setzen
., c218 99 e0 c1  sta $c1e0,y
., c21b c8        iny
., c21c 8c de c1  sty $c1de
., c21f a2 47     ldx #$47
., c221 20 87 c2  jsr $c287     Frage nach Disk-ID
., c224 a2 00     ldx #$00
., c226 20 cf ff  jsr $ffcf     Eingabe abwarten
., c229 c9 0d     cmp #$0d
., c22b f0 09     beq $c236
., c22d 99 e0 c1  sta $c1e0,y   ID ebenfalls abspeichern
., c230 c8        iny
., c231 e8        inx
., c232 e0 02     cpx #$02
., c234 90 f0     bcc $c226
., c236 8c df c1  sty $c1df
., c239 a2 53     ldx #$53
., c23b 20 87 c2  jsr $c287     'FROM TRACK:$' ausgeben
., c23e 20 cf ff  jsr $ffcf
., c241 85 fa     sta $fa
., c243 20 cf ff  jsr $ffcf
., c246 85 fb     sta $fb
., c248 a9 00     lda #$00
., c24a 85 d0     sta $d0
., c24c a2 62     ldx #$62
., c24e 20 87 c2  jsr $c287     'TO TRACK:$' ausgeben
., c251 20 cf ff  jsr $ffcf
., c254 85 fc     sta $fc
., c256 20 cf ff  jsr $ffcf
., c259 85 fd     sta $fd
., c25b a9 00     lda #$00
., c25d 85 d0     sta $d0
., c25f a5 fa     lda $fa
., c261 a6 fb     ldx $fb
., c263 20 04 c4  jsr $c404     Umrechnung in HEX-Byte
., c266 8d dc c1  sta $c1dc     Anfangstrack setzen
., c269 a5 fc     lda $fc
., c26b a6 fd     ldx $fd
., c26d 20 04 c4  jsr $c404     Umrechnung in HEX-Byte
., c270 8d dd c1  sta $c1dd     Endetrack setzen
., c273 ee dd c1  inc $c1dd     plus 1 als Vergleichswert
., c276 ea        nop
., c277 ea        nop
., c278 ea        nop
., c279 ea        nop
., c27a ea        nop
., c27b ea        nop
., c27c ea        nop
., c27d ea        nop
., c27e ea        nop
., c27f ea        nop
., c280 ea        nop
., c281 ea        nop
., c282 ea        nop
., c283 ea        nop
., c284 4c 93 c2  jmp $c293     weiter

., c287 bd 4d c3  lda $c34d,x   Ausgabe der Texte
., c28a f0 06     beq $c292
., c28c 20 d2 ff  jsr $ffd2
., c28f e8        inx
., c290 d0 f5     bne $c287
., c292 60        rts

., c293 a9 0d     lda #$0d
., c295 20 d2 ff  jsr $ffd2
., c298 a9 0d     lda #$0d
., c29a 20 d2 ff  jsr $ffd2
., c29d a9 00     lda #$00
., c29f a2 c0     ldx #$c0
., c2a1 85 a7     sta $a7
., c2a3 86 a8     stx $a8
., c2a5 a9 00     lda #$00
., c2a7 a2 05     ldx #$05
., c2a9 85 a9     sta $a9
., c2ab 86 aa     stx $aa
., c2ad a9 08     lda #$08      LISTEN für Gerät Nummer 8
., c2af 20 b1 ff  jsr $ffb1
., c2b2 a9 6f     lda #$6f      15;' Kommandokanal
., c2b4 20 93 ff  jsr $ff93
., c2b7 a9 4d     lda #$4d
., c2b9 20 a8 ff  jsr $ffa8     Programm zur Floppy senden
., c2bc a9 2d     lda #$2d
., c2be 20 a8 ff  jsr $ffa8
., c2c1 a9 57     lda #$57
., c2c3 20 a8 ff  jsr $ffa8
., c2c6 a0 00     ldy #$00
., c2c8 a5 a9     lda $a9
., c2ca 20 a8 ff  jsr $ffa8
., c2cd a5 aa     lda $aa
., c2cf 20 a8 ff  jsr $ffa8
., c2d2 a9 1e     lda #$1e
., c2d4 20 a8 ff  jsr $ffa8
., c2d7 b1 a7     lda ($a7),y
., c2d9 20 a8 ff  jsr $ffa8
., c2dc c8        iny
., c2dd c0 1e     cpy #$1e
., c2df 90 f6     bcc $c2d7
., c2e1 20 ae ff  jsr $ffae
., c2e4 18        clc
., c2e5 a5 a7     lda $a7
., c2e7 69 1e     adc #$1e
., c2e9 85 a7     sta $a7
., c2eb 90 03     bcc $c2f0
., c2ed e6 a8     inc $a8
., c2ef 18        clc
., c2f0 a5 a9     lda $a9
., c2f2 a6 aa     ldx $aa
., c2f4 69 1e     adc #$1e
., c2f6 85 a9     sta $a9
., c2f8 90 02     bcc $c2fc
., c2fa e6 aa     inc $aa
., c2fc e0 07     cpx #$07
., c2fe 90 ad     bcc $c2ad
., c300 c9 00     cmp #$00
., c302 90 a9     bcc $c2ad
., c304 a9 08     lda #$08      LISTEN für Gerät 8
., c306 20 b1 ff  jsr $ffb1
., c309 a9 6f     lda #$6f      15; Kommandokanal
., c30b 20 93 ff  jsr $ff93
., c30e a9 4d     lda #$4d
., c310 20 a8 ff  jsr $ffa8
., c313 a9 2d     lda #$2d
., c315 20 a8 ff  jsr $ffa8     Programm in der Floppy
., c318 a9 45     lda #$45      bei Adresse $0600 starten
., c31a 20 a8 ff  jsr $ffa8
., c31d a9 60     lda #$60
., c31f 20 a8 ff  jsr $ffa8
., c322 a9 06     lda #$06
., c324 20 a8 ff  jsr $ffa8
., c327 20 ae ff  jsr $ffae
., c32a a9 00     lda #$00
., c32c 85 90     sta $90
., c32e a9 08     lda #$08      Fehlermeldung holen
., c330 20 b4 ff  jsr $ffb4
., c333 a9 6f     lda #$6f
., c335 20 96 ff  jsr $ff96
., c338 20 a5 ff  jsr $ffa5
., c33b 20 d2 ff  jsr $ffd2     und anzeigen
., c33e 24 90     bit $90
., c340 50 f6     bvc $c338
., c342 20 ab ff  jsr $ffab
., c345 4c dc c3  jmp $c3dc     Endebehandlung
.: C348 00 00 00 00 00 93 20 20
.: C350 20 20 20 20 20 2A 2A 2A
.: C358 20 44 49 53 4B 2D 46 4F
.: C360 52 4D 41 54 2D 53 59 53
.: C368 54 45 4D 20 2A 2A 2A 0D
.: C370 0D 0D 20 28 43 29 20 31
.: C378 39 38 35 20 42 59 20 4B
.: C380 4F 53 53 20 20 20 0D 0D
.: C388 0D 44 49 53 4B 4E 41 4D
.: C390 45 3A 20 00 0D 0D 44 49
.: C398 53 4B 2D 49 44 3A 20 00
.: C3A0 0D 0D 46 52 4F 4D 20 54
.: C3A8 52 41 43 4B 3A 24 00 0D
.: C3B0 0D 54 4F 20 54 52 41 43
.: C3B8 4B 3A 24 00 0D 0D 41 4E
.: C3C0 4F 54 48 45 52 20 46 4F
.: C3C8 52 4D 41 54 20 28 59 2F
.: C3D0 4E 29 20 3F 20 0D 0D 00
.: C3D8 00 00 00 00 20 29 C4 A2
., c3dc 20 29 c4  jsr $c429     SAVE-Vektor stellen
., c3df a2 6f     ldx #$6f
., c3e1 20 87 c2  jsr $c287     'ANOTHER FORMAT (Y/N) ?' ausgeben
., c3e4 20 e4 ff  jsr $ffe4
., c3e7 f0 fb     beq $c3e4
., c3e9 c9 59     cmp #$59      Ende ?
., c3eb d0 03     bne $c3f0     ja: RTS
., c3ed 4c 00 c2  jmp $c200     noch einmal formatieren
., c3f0 60        rts
., c3f1 00        brk
., c3f2 a5 b7     lda $b7       Länge des Filenamens =0?
., c3f4 f0 03     beq $c3f9     ja; dann formatieren
., c3f6 4c ed f5  jmp $f5ed     zur SAVE-Routine
., c3f9 20 00 c2  jsr $c200     formatieren
., c3fc a9 01     lda #$01
., c3fe a2 00     ldx #$00
., c400 a0 00     ldy #$00
., c402 18        clc
., c403 60        rts           Ende

., c404 85 02     sta $02       Umrechnung der Eingabe in ein
., c406 86 03     stx $03       HEX-Byte
., c408 a5 02     lda $02
., c40a c9 41     cmp #$41
., c40c 90 03     bcc $c411
., c40e 18        clc
., c40f 69 09     adc #$09
., c411 29 0f     and #$0f
., c413 0a        asl
., c414 0a        asl
., c415 0a        asl
., c416 0a        asl
., c417 85 02     sta $02
., c419 a5 03     lda $03
., c41b c9 41     cmp #$41
., c41d 90 03     bcc $c422
., c41f 18        clc
., c420 69 09     adc #$09
., c422 29 0f     and #$0f
., c424 05 02     ora $02
., c426 85 02     sta $02
., c428 60        rts

., c429 a9 f2     lda #$f2      SAVE-Vektor herstellen
., c42b 8d 32 03  sta $0332
., c22e a9 c3     lda #$c3      auf Adresse $c3f2 setzen
., c230 8d 33 03  sta $0333
., c233 60        rts
```

Listing 21. (Schluß). Zwischen den Adressen C348 und C3DA liegt eine ASCII-Tabelle.

```
10 REM ************************     <137>
20 REM *                      *     <247>
30 REM *  DISK-FORMAT-SYSTEM  *     <052>
40 REM *                      *     <011>
50 REM *   (C) 1985 BY KOSS   *     <091>
60 REM *                      *     <031>
70 REM ************************     <197>
80 DATA 5657,5638,6947,7770,8264,7062,8578
   ,6111,3989,3215,9192,10797       <224>
90 DATA 8104,8232,8308,3524,3180,5204,4577 <144>
100 DATA 0,14,8,10,0,158,32,50,48,54,52,32
    ,32,0,0,0,162,64,160,8,134,2,132,3  <156>
110 DATA 162,0,160,192,134,4,132,5,160,0,1
    62,5,177,2,145,4,200,208,249,230,3  <187>
120 DATA 230,5,202,208,242,120,169,242,141
    ,50,3,169,195,141,51,3,88,96,234,234  <071>
130 DATA 165,10,201,36,144,7,169,18,133,67
    ,76,19,5,32,75,242,133,67,169,0,133  <043>
140 DATA 27,160,0,162,0,165,57,153,0,3,200
    ,200,165,27,153,0,3,200,165,10,153  <201>
150 DATA 0,3,200,165,19,153,0,3,200,165,18
    ,153,0,3,200,169,15,153,0,3,200,153  <254>
160 DATA 0,3,200,169,0,89,250,2,89,251,2,8
    9,252,2,89,253,2,153,249,2,230,27  <218>
170 DATA 165,27,197,67,144,190,169,3,133,4
    9,152,72,138,157,0,7,232,208,250,32  <092>
180 DATA 48,254,104,168,136,32,229,253,32,
    245,253,169,7,133,49,32,233,245,133  <100>
190 DATA 58,32,143,247,169,0,133,50,32,14,
    254,169,255,141,1,28,162,5,80,254  <251>
200 DATA 184,202,208,250,162,10,164,50,80,
    254,184,185,0,3,141,1,28,200,202,208  <125>
210 DATA 243,162,9,80,254,184,169,85,141,1
    ,28,202,208,245,169,255,162,5,80,254  <184>
220 DATA 184,141,1,28,202,208,247,162,187,
    80,254,184,189,0,1,141,1,28,232,208  <122>
230 DATA 244,160,0,80,254,184,177,48,141,1
    ,28,200,208,245,169,85,162,8,80,254  <146>
240 DATA 184,141,1,28,202,208,247,165,50,2
    4,105,10,133,50,198,27,208,149,80  <041>
250 DATA 254,184,80,254,184,32,0,254,169,2
    00,133,31,169,0,133,48,169,3,133,49  <160>
260 DATA 165,67,133,27,32,86,245,162,10,16
    0,0,80,254,184,173,1,28,209,48,208  <123>
270 DATA 14,200,202,208,242,24,165,48,105,
    10,133,48,76,53,6,198,31,208,209,169  <225>
280 DATA 6,76,211,253,32,86,245,160,187,80
    ,254,184,173,1,28,217,0,1,208,231  <087>
290 DATA 200,208,242,162,252,80,254,184,17
    3,1,28,217,0,7,208,215,200,202,208  <125>
300 DATA 241,198,27,208,176,76,158,253,160
    ,0,185,224,6,153,0,2,200,204,223,6  <150>
310 DATA 144,244,173,223,6,141,116,2,173,2
    22,6,141,123,2,169,0,133,127,32,0  <084>
320 DATA 193,172,123,2,185,0,2,133,18,185,
    1,2,133,19,32,7,211,169,26,141,5,28  <206>
330 DATA 169,192,133,0,165,0,48,252,174,22
    0,6,134,10,169,224,133,2,165,2,48  <128>
340 DATA 252,201,2,176,12,232,236,221,6,14
    4,236,32,64,238,96,234,234,162,2,76  <241>
350 DATA 10,230,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  <204>
360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  <248>
370 DATA 162,0,32,135,194,160,0,32,207,255
    ,201,13,240,8,153,224,193,200,192  <149>
380 DATA 16,144,241,169,44,153,224,193,200
    ,140,222,193,162,71,32,135,194,162  <233>
390 DATA 0,32,207,255,201,13,240,9,153,224
    ,193,200,232,224,2,144,240,140,223  <206>
400 DATA 193,162,83,32,135,194,32,207,255,
    133,250,32,207,255,133,251,169,0,133  <097>
410 DATA 208,162,98,32,135,194,32,207,255,
    133,252,32,207,255,133,253,169,0,133  <114>
420 DATA 208,165,250,166,251,32,4,196,141,
    220,193,165,252,166,253,32,4,196,141  <125>
430 DATA 221,193,238,221,193,234,234,234,2
    34,234,234,234,234,234,234,234,234  <031>
440 DATA 234,234,76,147,194,189,77,195,240
    ,6,32,210,255,232,208,245,96,169,13  <123>
450 DATA 32,210,255,169,13,32,210,255,169,
    0,162,192,133,167,134,168,169,0,162  <100>
460 DATA 5,133,169,134,170,169,8,32,177,25
    5,169,111,32,147,255,169,77,32,168  <094>
470 DATA 255,169,45,32,168,255,169,87,32,1
    68,255,160,0,165,169,32,168,255,165  <165>
480 DATA 170,32,168,255,169,30,32,168,255,
    177,167,32,168,255,200,192,30,144  <050>
490 DATA 246,32,174,255,24,165,167,105,30,
    133,167,144,3,230,168,24,165,169,166  <204>
500 DATA 170,105,30,133,169,144,2,230,170,
    224,7,144,173,201,0,144,169,169,8  <039>
510 DATA 32,177,255,169,111,32,147,255,169
    ,77,32,168,255,169,45,32,168,255,169  <001>
520 DATA 69,32,168,255,169,96,32,168,255,1
    69,6,32,168,255,32,174,255,169,0,133  <009>
530 DATA 144,169,8,32,180,255,169,111,32,1
    50,255,32,165,255,32,210,255,36,144  <183>
540 DATA 80,246,32,171,255,76,220,195,0,0,
    0,0,0,147,32,32,32,32,32,32,32,42  <048>
550 DATA 42,42,32,68,73,83,75,45,70,79,82,
    77,65,84,45,83,89,83,84,69,77,32,42  <006>
560 DATA 42,42,13,13,13,32,40,67,41,32,49,
    57,56,53,32,66,89,32,75,79,83,83,32  <223>
570 DATA 32,32,13,13,13,68,73,83,75,78,65,
    77,69,58,32,0,13,13,68,73,83,75,45  <197>
580 DATA 73,68,58,32,0,13,13,70,82,79,77,3
    2,84,82,65,67,75,58,36,0,13,13,84  <155>
590 DATA 79,32,84,82,65,67,75,58,36,0,13,1
    3,65,78,79,84,72,69,82,32,70,79,82  <246>
600 DATA 77,65,84,32,40,89,47,78,41,32,63,
    32,13,13,0,0,0,0,0,32,41,196,162,111  <016>
610 DATA 32,135,194,32,228,255,240,251,201
    ,89,208,3,76,0,194,96,0,165,183,240  <011>
620 DATA 3,76,237,245,32,0,194,169,1,162,0
    ,160,0,24,96,133,2,134,3,165,2,201  <194>
630 DATA 65,144,3,24,105,9,41,15,10,10,10,
    10,133,2,165,3,201,65,144,3,24,105  <176>
640 DATA [illegible],41,15,5,2,133,2,96,169,242,141,
    50,3,169,195,141,51,3,96,0  <095>
1000 REM                            <122>
1010 REM **** DATAS INITIALISIEREN  <178>
1020 REM                            <142>
1030 RESTORE:PRINT:PRINT:PRINT"DATAS WERDE
     N UEBERPRUEFT !!!":PRINT:PRINT  <215>
1040 CLR:DIM P(19):DIM W(19)        <077>
1050 FOR X=0 TO 18:READ P(X):P=P+P(X):NEXT <139>
1060 IF P<>124349 THEN PRINT"PRUEFSUMMENFE
     HLER":PRINT:PRINT:LIST 80-90    <038>
1070 FOR X=0 TO 18:FOR Y=0 TO 59:READ A:W(
     X)=W(X)+A:NEXT Y               <240>
1080 IF W(X)<>P(X)THEN 1150         <146>
1090 NEXT X                         <032>
1100 PRINT:PRINT"DIE DATAS SIND OK UND WER
     DEN":PRINT:PRINT"ABGESPEICHERT!"  <116>
1110 RESTORE:FOR X=0 TO 18:READ A:NEXT  <166>
1120 FOR X=0 TO 1139:READ A:POKE X+2048,A:
     NEXT                           <093>
1130 POKE 45,119:POKE 174,119:POKE 46,12:P
     OKE 175,12:CLR                 <006>
1140 PRINT:PRINT"MIT 'SAVE' ABSPEICHERN!":
     PRINT:END                      <097>
1150 REM FEHLERBEHANDLUNG           <159>
1160 PRINT:PRINT"FEHLER IN DEN DATAS"X*60"
      BIS"X*60+59" !":Z=INT(X*600/17.8)  <161>
1170 PRINT:PRINT:PRINT"DAS ENTSPRICHT IN E
     TWA DEN{14SPACE,DOWN}ZEILEN AB"Z  <073>
1180 END                            <032>
```

Listing 22. Der DATA-Lader der Formatierroutine

kette eingeben (maximal 16 Zeichen werden angenommen). Danach erwartet der Computer eine zweistellige ID. Schließlich, und das ist das Besondere an diesem Programm, können Sie noch den ersten und letzten zu formatierenden Track eingeben. Diese Eingabe muß hexadezimal erfolgen und erlaubt einen Bereich von $01 bis $FF.

Achtung! Wird eine Zahl größer als $29 (41) eingegeben, wird es in der Regel kritisch. Der Kopf ist dann nämlich am oberen Anschlagpunkt angelangt.

Etwas ist noch zu beachten: Ein Nachformatieren einer Spur auf einer gefüllten Diskette ist mit dem Programm ohne Änderung nicht möglich, da das Directory auf jeden Fall neu geschrieben wird. Wird die Diskette nicht vollständig formatiert, so ist darauf zu achten, daß die gleiche ID eingegeben

wird, wie sie schon für die übrige Diskette Gültigkeit hat, da es sonst einen »29, DISK ID MISMATCH ERROR« gibt.

Wollen Sie dennoch einen Einzeltrack neu formatieren, ohne das Directory zu zerstören, so können Sie das durch eine einfache Änderung im Floppy-Programm erreichen. Sie gehen in Listing 21 an die Adresse $06B5. Den Befehl JSR $EE40 und das nachfolgende RTS ersetzen Sie durch lauter NOPs.

Eine Änderung des Directory-Track unterbleibt jetzt, sofern Sie die Track-Nummern zur Formatierung entsprechend wählen, da dieser Befehl die Routine zum kurzen Formatieren im DOS aufgerufen hätte.

In jedem Fall gilt aber: Bei Formatieren von Einzel-Tracks müssen diese die gleiche ID wie die übrige Diskette erhalten.

Eine weitere Möglichkeit dient der Schonung des Laufwerks. Wenn Sie sich das Floppy-Programm noch einmal betrachten, dann finden Sie bei Adresse $0696 den Befehl an den Diskcontroller, einen BUMP auszuführen. Wenn Sie hier das $C0 durch ein $00 ersetzen, dann unterbleibt dieses Anschlagen des Tonkopfes am Anfang des Formatierens. Diese Maßnahme ist immer dann nützlich, wenn mehrere Disketten hintereinander formatiert werden sollen.

Zur Zeitdauer ist noch zu sagen, daß das Programm für eine Diskette zirka 30 Sekunden benötigt und damit um einiges schneller ist als das Programm im DOS der Floppy 1541. Warum das so ist, sollen wir gleich erfahren.

Geschwindigkeit; aber wie?

In meinem Formatierprogramm wurde die Berechnung der Lücke zwischen zwei Sektoren weggelassen. Wir können nämlich davon ausgehen, daß diese Lücken auf jeder Diskette in etwa gleich sind. Aus diesem Grund verwende ich einfach einen Erfahrungswert für die Länge der Lücke, der zusätzlich noch einen Sicherheitsbereich enthält. Diesen Wert sehen Sie in Listing 21 an der Adresse $05DF.

Wenn Sie mit dem Programm Disketten formatieren, werden Sie feststellen, daß die Datensicherheit auch weiterhin voll gewährleistet ist.

Im Gegensatz zu anderen schnellen Formatierprogrammen wurde aber nicht auf ein Verify verzichtet, da das Formatieren die einzige Möglichkeit bietet, defekte Disketten rechtzeitig zu erkennen, ohne daß dabei wichtige Daten verlorengehen. Einmal ganz davon abgesehen, macht das Verifizieren außerdem nur einen sehr kleinen Teil am Geschwindigkeitsverlust aus, so daß die Sicherheit vor einigen Sekunden Zeitgewinn Vorrang haben sollte.

Wollen Sie die Zeit dennoch einmal ohne Verify messen, so »klemmen« sie den Rest der Formatierungsroutine ab $05FD ganz einfach ab, indem Sie an dieser Stelle nach JSR $FE00 ein JMP $FD9E einfügen.

Eine weitere Verbesserung gegenüber dem DOS 2.6 der Floppy 1541 hat eigentlich mehr kosmetischen Charakter. Es geht hier um den Leerinhalt von Datenblöcken, nachdem eine Diskette neu formatiert wurde. Den Inhalt werden Sie höchstwahrscheinlich schon kennen: Es steht am Anfang des Datenblocks ein $4B gefolgt von 255 $01-Byte.

Dieser Inhalt ist eigentlich auf einen Fehler im DOS zurückzuführen; er müßte, wie auch bei den großen Commodore-Diskettenlaufwerken aus 256 $00-Byte bestehen.

Im Programm werden alle Sektoren mit dem üblichen Wert $00 gefüllt. Noch ein paar Hinweise zur Benutzung des Formatierprogramms.

Nach RUN wird automatisch der SAVE-Vektor auf den Programmstart der Formatierroutine gestellt. Wird kein Filename angegeben, so erfolgt ein Sprung in das Formatierprogramm. Durch Drücken von <RUN/STOP+RESTORE> läßt sich der SAVE-Vektor wieder richtig »hinbiegen«. Hierzu dürfte jedoch kein Anlaß bestehen, da ansonsten bei fehlendem Filenamen kein Programm gestartet wird.

Mußten Sie dennoch einmal <RESTORE> drücken, so läßt sich das Formatiersystem mit SYS 49664 ($C200) erneut starten; nach Beendigung wird unter anderem auch der SAVE-Vektor wieder auf das Programm zurückgestellt.

Wollen Sie sich den Disk-Status anzeigen lassen, so tippen Sie SYS49962. Es erscheint danach auch die Frage nach einem weiteren Formatiervorgang, die Sie entsprechend beantworten. Nach dieser Anzeige wird der SAVE-Vektor ebenfalls wieder hergestellt.

Ich möchte Sie an dieser Stelle auf ein paar Speicherstellen in der Zero-Page der 1541 aufmerksam machen. Wie Sie wissen, werden dort nach einem Reset ein paar Konstanten abgelegt, die vom Benutzer (beliebig) verändert werden können. Mit den Konstanten meine ich zum Beispiel $08 als Kennzeichen eines Blockheaders oder $07 als Kennzeichen eines Datenblocks.

Wie Sie aus der Zero-Page-Belegung entnehmen können, werden diese beiden Werte in den Speicherstellen $47 (Wert 07) und $39 (Wert 08) abgelegt und können nun abgeändert werden. Der neue Wert, den Sie vielleicht in diese Speicherstellen eintragen, sollte sich jedoch im Bereich von $00 bis $0F bewegen, da es sonst Schwierigkeiten beim Lesen geben kann. Die Folge eines Leseversuchs mit normalen Werten, wenn eine Diskette anders formatiert wurde, sind entweder ein »20, READ ERROR« oder ein »22, READ ERROR«.

Der Vorteil dieser Errors ist jedoch die Möglichkeit, den Blöcken auch Inhalte mitzugeben, womit ein sehr wirkungsvoller Kopierschutz konstruiert werden kann.

Zum Lesen oder Beschreiben der Diskette müssen die Werte in den beiden Speicherstellen nur jeweils richtig gestellt werden; dann kann ein ganz normaler Zugriff stattfinden.

Mit Hilfe des Formatierprogramms können Sie jetzt auch noch zusätzlich illegale Spuren beschreiben. Hierbei müssen Sie allerdings, wie vorhin besprochen, auf Job-Schleifenebene arbeiten, um die Begrenzung auf die Spuren 1 bis 35 zu umgehen.

Was ist eine GCR-Codierung?

Vielleicht sind Ihnen bestimmt schon einige Ungereimtheiten aufgefallen, was den Direktzugriff auf die Diskette betrifft. Auch im Abschnitt über das Formatieren waren zum Beispiel im Listing von S-Format einige Sprungbefehle, die nicht erklärt wurden.

Erinnern Sie sich noch an den Abschnitt, der sich das erste Mal mit dem Schreiben von Daten auf die Diskette beschäftigte? Dort wurden unter anderem die SYNC-Markierungen auf der Diskette besprochen, die dem Diskcontroller als Positionsanzeiger dienen.

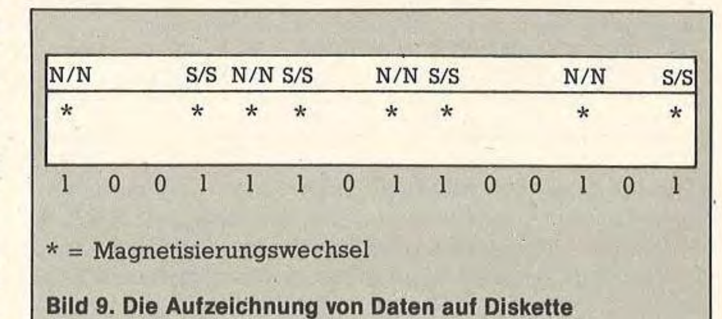


Bild 9. Die Aufzeichnung von Daten auf Diskette (schematisch)

Es wurde darin erwähnt, daß sich diese SYNC-Markierungen bei der Floppy 1541 aus 5 $FF-Byte zusammensetzen, die hintereinander auf Diskette geschrieben werden. Was ist aber, wenn ein Datenblock geschrieben werden soll, der nur aus $FF-Bytes besteht? Eigentlich müßten dann diese Bytes als SYNC-Markierung wirksam werden und den gesamten Schreib- und Lesebetrieb stören. Wie die Praxis zeigt, tritt dieser Fehler nicht auf. Auch bei mehreren Blöcken aus $FF-Bytes kommt es zu keinen Komplikationen. Bei der Konstruktion der Floppystation hat man sich nämlich eine Codierung der Daten einfallen lassen, die eine Eindeutigkeit der Daten schafft. Die Codierung heißt GCR, was nichts anderes als eine Abkürzung der englischen Wörter »Group Code Recording« ist.

Es stellt sich jetzt natürlich die Frage, was bei der GCR-Codierung passiert, damit eine Verwechslung zwischen SYNC- und Daten-Bytes unmöglich wird. Zur Beantwortung dieser Frage muß ein wenig intensiver auf das Lesen und Schreiben der Floppystation eingegangen werden.

Was macht die GCR-Codierung?

Das Lesen von Bytes durch den Lesekopf steuert ein Timer des Diskcontrollers. Auf der Diskette selbst wird jedes 1-Bit physikalisch durch einen Wechsel der Magnetisierungsrichtung dargestellt und ein 0-Bit durch gleichbleibende Richtung der Magnetisierung. Bild 9 zeigt, was gemeint ist. Soll ein Byte von Diskette gelesen werden, so wartet der Diskcontroller einfach die Zeitspanne ab, die zum Lesen von 8 Bit erforderlich ist. Innerhalb dieser Zeit liest der Schreib-/Lesekopf eine gewisse Folge von Magnetisierungs- und Nicht-Magnetisierungswechseln.

Dazu ein Beispiel: Auf der Diskette steht ein Byte mit dem Inhalt $55. $55 wird binär durch die Kombination %01010101 dargestellt. Der Tonkopf stellt also während der Lesezeit die folgenden Magnetisierungswechsel fest:

Magnetisierung wechselt nicht, wechselt, wechselt nicht, wechselt, wechselt nicht, wechselt, wechselt nicht, wechselt.

Das Erkennen eines Bits geschieht dabei völlig zeitgesteuert. Der Diskcontroller »weiß«, daß er zum Lesen eines Bits eine bestimmte Zeit warten muß. Danach gilt das Bit als gelesen, und es wird eine »1« oder eine »0« bereitgestellt, je nachdem, ob ein Magnetisierungswechsel stattgefunden hat oder nicht.

Hexadezimal	Binär	GCR
$0	0000	01010
$1	0001	01011
$2	0010	10010
$3	0011	10011
$4	0100	01110
$5	0101	01111
$6	0110	10110
$7	0111	10111
$8	1000	01001
$9	1001	11001
$A	1010	11010
$B	1011	11011
$C	1100	01101
$D	1101	11101
$E	1110	11110
$F	1111	10101

Tabelle 17. Umrechnungstabelle für Binär-GCR-Umwandlung

Praktisch könnte man das folgendermaßen beschreiben: Sie machen mit einem Freund eine Zeit von 10 Sekunden aus. Er hat dann die Aufgabe, innerhalb dieser 10 Sekunden entweder zu pfeifen oder nicht. Danach warten Sie diese 10 Sekunden ab. Hat er während dieser Zeit gepfiffen, dann entspricht das einem Magnetisierungswechsel. Hat er innerhalb der 10 Sekunden nicht gepfiffen, bedeutet das ein »0«-Bit, also keinen Magnetisierungswechsel. Da eine Diskette im Laufwerk nicht absolut gleichmäßig gedreht werden kann, also Drehzahlschwankungen unterliegt, muß noch für eine Kompensation der mechanischen Fehler gesorgt werden. Dazu wird der Timer, der die abzuwartende Zeit für jedes Bit bestimmt, bei jedem Magnetisierungswechsel neu getriggert (gestellt). Ein »1«-Bit hat also neben seinem Informationsgehalt noch die wichtige Aufgabe, Laufwerksschwankungen auszugleichen, um Lesefehler zu verhindern. Aus diesem Grund darf es zum Beipiel nicht passieren, daß mehrere $00-Bytes hintereinander auf der Diskette stehen, da sonst zu lange keine Laufwerkskontrolle mehr stattfinden könnte.

Aber auch zuviele »1«-Bits sind nicht gestattet, da mehr als acht »1«-Bits ein SYNC-Signal auslösen.

Aus den genannten Gründen werden alle Daten, die auf die Diskette geschrieben werden, vorher GCR-codiert. Mit dieser Codierung wird ausgeschlossen, daß mehr als acht »1«-Bit und mehr als zwei »0«-Bit direkt hintereinander auf die Diskette geschrieben werden und so die Schreib- und Lese-Elektronik durcheinanderbringen.

Einzig und allein die SYNC-Markierungen (mehr als acht »1«-Bit) werden vom DOS (Disk Operating System, Controller) uncodiert auf die Diskette geschrieben.

Es gibt zwei Schreibarten

Man kann also zwischen zwei Schreibarten auf Diskette unterscheiden:

1) Schreiben von Markierungen.
Hier werden fünf $FF-Byte direkt hintereinander auf die Diskette geschrieben, um eine SYNC-Markierung zu bilden, die der Orientierung dient.

2) Schreiben von Daten.
In diesem Modus werden Byte-Inhalte codiert, um sich von den Markierungen zu unterscheiden.

Sehen Sie sich jetzt einmal Tabelle 17 an, die Umwandlungstabelle für die Konvertierung Binär nach GCR und umgekehrt.

Wie Sie unschwer erkennen können, handelt es sich beim GCR-Code um einen 5-Bit-Code. Jedes 4-Bit-Nibble, das Sie umwandeln, wird zu einem 5-Bit-GCR-Nibble. Ein Byte, das vorher aus 8 Bit bestand, wird also durch die Codierung 10 Bit lang. Allgemein nimmt die Länge der codierten Daten um den Faktor 5/4 zu. Deshalb ist die Handhabung der GCR-Bytes nicht ganz einfach. Wandeln Sie doch einmal zwei Byte in den GCR-Code um. Als Ergebnis erhalten Sie »zweieinhalb« Byte, die sicherlich schwer zu behandeln sind.

Bei der GCR-Codierung geht man aus diesem Grund einen ganz einfachen Weg, um keine Formatprobleme zu bekommen: es werden jeweils immer 4 Byte gleichzeitig umgewandelt. Als Ergebnis erhält man 5 vollständige Byte, die ohne Probleme weiterverarbeitet werden können.

Lassen Sie mich das einmal an einem Beipiel erläutern:

Nehmen wir einmal an, wir hätten vier Byte mit dem Wert $FF. Eine Kombination also, die nicht direkt auf die Diskette geschrieben werden darf.

Wir wandeln diese vier Hex-Byte nun in die entsprechenden fünf GCR-Byte um, indem wir in Tabelle 17 nachsehen, was die entsprechenden GCR-Äquivalente dieser Bytes sind. Wir kommen zu folgendem Ergebnis:

HEX	BINÄR	GCR-Code
$FF	1111 1111	10101 10101

$FF	1111 1111	10101 10101
$FF	1111 1111	10101 10101
$FF	1111 1111	10101 10101

Die binär dargestellten GCR-Werte müssen wir jetzt nur noch zu fünf Byte zusammenfassen, um auf folgendes Ergebnis zu kommen:

1010 + 1101 = AD
(1010 1+101 01)
0110 + 1011 = 6B
0101 + 1010 = 5A
1101 + 0110 = D6
1011 + 0101 = B5

Vier $FF-Byte werden also bei der GCR-Codierung in die fünf Byte $AD, $6B, $5A, $D6 und $B5 umgewandelt. Sie können sich jetzt leicht davon überzeugen, daß diese fünf Byte für den Diskcontroller absolut ungefährlich und unkritisch sind, und daß sie die vorgeschriebenen Normen (nicht mehr als zwei »0«-Byte und nicht mehr als acht »1«-Byte) erfüllen.

»Bit-Knotereien«

Um Ihnen die Umwandlung der Bytes zu erleichtern, habe ich diesem Kurs zwei Programmlistings beigefügt. Listing 23 enthält ein Programm, das Ihnen vier Hex-Byte in fünf GCR-Byte umwandelt. In Listing 24 sehen Sie ein Programm abgedruckt, das die GCR-Codierung wieder rückgängig macht. Hier werden fünf GCR-Byte in vier Hex-Bytes zurückverwandelt, wobei Sie mit unerlaubten Bitkombinationen vorsichtig sein sollten. Kann ein Byte nicht zurückverwandelt werden, so haben Sie eine unerlaubte GCR-Bitkombination, die sich im Ergebnis dadurch äußert, daß entsprechende Nibbles fehlen. Sie erhalten dann unter Umständen nur »halbe« Bytes.

```
10 REM PROGRAMM ZUR KONVERTIERUNG               <242>
20 REM VON VIER HEXBYTES IN DIE                 <161>
30 REM FUENF ENTSPRECHENDEN                     <055>
40 REM GCR-AEQUIVALENTE                         <068>
50 REM                                          <193>
60 REM                                          <203>
70 REM                                          <213>
80 REM (W) 1985 BY KARSTEN SCHRAMM              <028>
90 REM                                          <233>
100 A$="0123456789ABCDEF":DIM G$(15):E$="" <220>
110 G$(0)="01010"                               <066>
120 G$(1)="01011"                               <078>
130 G$(2)="10010"                               <088>
140 G$(3)="10011"                               <100>
150 G$(4)="01110"                               <111>
160 G$(5)="01111"                               <123>
170 G$(6)="10110"                               <133>
180 G$(7)="10111"                               <145>
190 G$(8)="01001"                               <154>
200 G$(9)="11001"                               <166>
210 G$(10)="11010"                              <216>
220 G$(11)="11011"                              <228>
230 G$(12)="01101"                              <238>
240 G$(13)="11101"                              <250>
250 G$(14)="11110"                              <005>
260 G$(15)="10101"                              <016>
270 PRINT"{CLR}HEX - GCR - KONVERTIERUNG":
    PRINT                                       <096>
280 PRINT:PRINT"GEBEN SIE JETZT 4 HEXBYTES
     EIN":PRINT                                 <235>
290 PRINT"Z.B. ED 34 27 58":INPUT"{2DOWN}"
    ;H$:GC$=""                                  <104>
300 GOSUB 470:FOR X=1 TO 4                      <035>
310 H1$=MID$(H$,X*2-1,1):H2$=MID$(H$,X*2,1
    )                                           <120>
320 H1=VAL(H1$):H2=VAL(H2$)                     <057>
330 IF H1=0 AND H1$<>"0"THEN H1=ASC(H1$)-5
    5                                           <240>
340 IF H2=0 AND H2$<>"0"THEN H2=ASC(H2$)-5
    5                                           <254>
350 GC$=GC$+G$(H1)+G$(H2)                       <044>
360 NEXT X                                      <067>
370 FOR X=1 TO 10                               <052>
380 B=0:B$=MID$(GC$,X*4-3,4)                    <094>
390 FOR Y=0 TO 3                                <026>
400 IF MID$(B$,Y+1,1)="1"THEN B=B+2↑(3-Y)  <112>
410 NEXT Y                                      <118>
420 E$=E$+MID$(A$,B+1,1)                        <249>
430 IF X/2=INT(X/2)THEN E$=E$+" "               <121>
440 NEXT X                                      <147>
450 PRINT:PRINT:PRINT"GCR: ";E$                 <000>
460 END                                         <077>
470 X$="":FOR X=1 TO LEN(H$)                    <099>
480 IF MID$(H$,X,1)<>" "THEN X$=X$+MID$(H$
    ,X,1)                                       <223>
490 NEXT                                        <109>
500 H$=X$:RETURN                                <087>
```

Listing 23. Umwandlung von Daten in GCR-Bytes

Die Floppystation hält übrigens für diesen Fall eine Fehlermeldung bereit, einen »24, READ ERROR«.

Im DOS existieren übrigens die folgenden Routinen zur Konvertierung:

$F6D0: Dieses Programm holt vier Hex-Byte aus den Speicherstellen $52 bis $55 und wandelt diese Bytes in die fünf entsprechenden GCR-Werte um. Diese fünf Byte werden anschließend im Puffer der Adresse $30/31 (L,H) mit dem Pufferzeiger in $34 abgelegt.

Pufferadresse und Pufferzeiger müssen dabei vor Aufruf dieser Routine übergeben werden.

$F78F: Diese Routine wandelt einen gesamten Puffer, dessen Adresse in $30/31 (L,H) stehen muß, in GCR-Werte um und speichert diese in den Ausweichpuffer sowie den ursprünglichen Puffer zurück. Der Pufferinhalt vergrößert sich durch diese Umwandlung von 256 auf 324 Bytes.

$F7E6: Diese Routine wandelt fünf GCR-Byte aus einem Puffer (dessen Adresse in $30/31 (L,H) und dessen Pufferzeiger in $34 steht) wieder in vier Hex-Byte zurück, wobei diese dann in der Zero-Page von $52 bis $55 gespeichert werden.

$F8E0: Diese Routine decodiert einen gesamten GCR-Pufferinhalt in die ursprüngliche Form und legt diese 256 Byte dann im Puffer mit der Adresse $30/31 (L,H) ab. Die vorherigen 324 GCR-Byte müssen im gleichen Puffer und im Ausweichpuffer ($01BB bis $01FF) stehen.

Die Anwendungen dieser Routinen sind äußerst vielfältig. So können Sie diese Programme zum Beispiel für einen Diskmonitor verwenden, in dem man zwischen der Anzeige von GCR-Bytes und der Anzeige von normalen Hex-Bytes hin- und herschalten kann. Die einzigen Änderungen, die Sie dazu machen müssen, bestehen in der Umrechnung der Adressen für die Speicherbereiche im Computer und der Angabe neuer Parameter als Puffer- und Zero-Page-Bereiche. Ihrer Phantasie, was die Möglichkeiten des Monitors angeht, sind außer dem Speicherplatz im Computer keine Grenzen gesetzt.

Bis zu 365 Byte in einem Block

Durch die Verwendung der GCR-Codierung ergeben sich noch Konsequenzen. Wie sieht es beispielsweise in den Puffern der Floppystation aus, wenn ein Puffer mit einem vollständigen Datenblock (also 256 Byte) gefüllt wurde und dieser aufgezeichnet werden soll? Für dieses Problem hat der Controller einen speziellen Ausweichpuffer. Der Puffer hat eine Größe von 68 Byte und befindet sich im Bereich von $01BB bis $01FF.

Wird nun ein Datenblock in Puffer 1 ($0400 bis $04FF) codiert, so werden die ersten 68 GCR-Byte in den Ausweichpuffer übernommen. Die restlichen Bytes stehen in Puffer 1.

Aus den 256 Byte an Information macht das DOS durch die Konvertierung also 324 Byte, die einen gesamten Daten-

```
10 REM PROGRAMM ZUR KONVERTIERUNG            <242>
20 REM VON FUENF GCR-BYTES IN DIE            <003>
30 REM VIER ENTSPRECHENDEN                   <249>
40 REM HEX-AEQUIVALENTE                      <077>
50 REM                                       <193>
60 REM                                       <203>
70 REM                                       <213>
80 REM (W) 1985 BY KARSTEN SCHRAMM           <028>
90 REM                                       <233>
100 A$="0123456789ABCDEF":DIM G$(15):E$="" <220>
110 G$(0)="01010"                            <066>
120 G$(1)="01011"                            <078>
130 G$(2)="10010"                            <088>
140 G$(3)="10011"                            <100>
150 G$(4)="01110"                            <111>
160 G$(5)="01111"                            <123>
170 G$(6)="10110"                            <133>
180 G$(7)="10111"                            <145>
190 G$(8)="01001"                            <154>
200 G$(9)="11001"                            <166>
210 G$(10)="11010"                           <216>
220 G$(11)="11011"                           <228>
230 G$(12)="01101"                           <238>
240 G$(13)="11101"                           <250>
250 G$(14)="11110"                           <005>
260 G$(15)="10101"                           <016>
270 PRINT"{CLR}GCR - HEX - KONVERTIERUNG":
    PRINT                                    <096>
280 PRINT:PRINT"GEBEN SIE JETZT 5 GCR-BYTE
    S EIN":PRINT                             <016>
290 INPUT"{2DOWN}";H$:GC$=""                 <147>
300 X$="":FOR X=1 TO LEN(H$)                 <185>
310 IF MID$(H$,X,1)<>" "THEN X$=X$+MID$(H$
    ,X,1)                                    <053>
320 NEXT                                     <195>
330 H$=X$                                    <229>
340 FOR X=1 TO 10                            <022>
350 X$=MID$(H$,X,1)                          <245>
360 XX=VAL(X$):IF XX=0 AND X$<>"0"THEN XX=
    ASC(X$)-55                               <104>
370 FOR Y=0 TO 3                             <006>
380 YY=INT(XX/2↑(3-Y)):XX=XX-YY*2↑(3-Y)     <105>
390 IF YY THEN GC$=GC$+"1":GOTO 410          <240>
400 GC$=GC$+"0"                              <189>
410 NEXT Y,X                                 <250>
420 HC$="":FOR X=1 TO 8                      <028>
430 X$=MID$(GC$,X*5-4,5)                     <075>
440 FOR Y=0 TO 15                            <127>
450 IF X$<>G$(Y)THEN NEXT Y                  <197>
460 :                                        <007>
470 HC$=HC$+MID$(A$,Y+1,1)                   <206>
480 IF INT(X/2)=X/2 THEN HC$=HC$+" "         <055>
490 NEXT X                                   <197>
500 PRINT:PRINT:PRINT"HEX: ";HC$             <129>
```

Listing 24. Umwandlung von GCR- in Daten-Bytes

block darstellen (inklusive Prüfsumme). Natürlich werden auch die Parameter im Datenblock-Header (ID, Track, Sektor, Prüfsumme und Kennzeichen) vor dem Schreiben auf die Diskette in GCR-Bytes umgewandelt, wobei der Blockheader dann mit den zwei Lücken-Byte auf eine Länge von zehn GCR-Byte anwächst, da der Header aus ursprünglich acht Hex-Werten besteht.

Das war's

Zusammenfassend besteht ein Sektor auf der Diskette aus den fünf Byte der ersten SYNC-Markierung; danach folgen die zehn Byte des Blockheaders. Vor der SYNC-Markierung des Datenblocks folgen jedoch noch neun $55-Byte, die der GCR-Norm entsprechen und direkt auf die Diskette geschrieben werden. Sie dienen als Pufferlücke, in der dem Diskcontroller Zeit bleibt, zwischen Schreiben und Lesen umzuschalten.

Nach den fünf Byte der SYNC-Markierung folgen die 324 Byte des Datenblocks inklusive dessen Prüfsumme und anschließend noch die Lücke zwischen zwei Sektoren, die erfahrungsgemäß zwischen acht und zwölf Byte lang ist. Wie Sie sehen, hat also so ein Sektor auf der Diskette die stattliche Länge von 361 bis 365 Byte.

Jetzt werden Ihnen bestimmt auch ein paar zweifelhafte JSR-Befehle im vorigen Abschnitt klar: bei dem Formatiersystem wird einmal ein Befehl JSR $FE30 und an anderer Stelle ein Befehl JSR $F78F ausgeführt. Diese Adressen sind die Einsprünge der Codierroutinen.

Vielleicht kommt Ihnen auch noch einmal die Herstellung eines Killertracks in Erinnerung. Hier wird ein gesamter Track direkt mit $FF-Bytes vollgeschrieben und stellt so eine »Riesen-SYNC-Markierung« dar. Da eine solche Bitfolge jedoch unzulässig ist, kommt die Lese-und Schreibelektronik der Floppystation völlig aus dem Konzept; der Controller »stürzt ab«.

Wenn Sie noch mehr über Ihre Floppy 1541, über schnelle Kopierprogramme und Kopierschutz-Methoden erfahren oder ein gut dokumentiertes DOS-Listing haben wollen, dann sollten Sie einmal in das M&T Floppybuch schauen.

Auch als Besitzer einer 1570/71 sollten Sie sich ein entsprechendes Buch zulegen, da diese beiden Laufwerke erheblich mehr können, als in unserem Kurs dargestellt wurde. Wir haben nur immer den 1541-Modus besprochen.

(ks)

ROCKUS

Verwalten wie die Profis

Das Verwalten großer Datenmengen mit dem Computer ist zweifellos eine reizvolle Sache. Bei der Programmierung in Basic hat man jedoch schnell die Grenzen der Verarbeitungsgeschwindigkeit erreicht. Wir zeigen Ihnen alles, was Sie beim Schreiben von Dateiverwaltungen in Maschinensprache wissen müssen.

Es gibt leider eine große Anzahl von Assemblerprogrammierern, die die tollsten Routinen schreiben, wie zum Beispiel Sortierroutinen oder Basic-Erweiterungen, die jedoch nicht wissen, wie von Maschinensprache aus auf Drucker oder Diskettenlaufwerk zugegriffen werden kann.

Zugegebenermaßen ist das »Datei-Handling« keine einfache Angelegenheit und selbst in einer höheren Programmiersprache wie Basic aufgrund der vielfältigen Möglichkeiten recht aufwendig (Datei auf Diskette oder Datasette zum Schreiben oder Lesen öffnen; relative oder sequentielle Datei öffnen und so weiter).

Wenn Sie jedoch bereits in Basic mit Dateien gearbeitet haben, wird Ihnen das Verständnis dieses Artikels keine besondere Mühe bereiten, da die erforderlichen Prozeduren (Datei öffnen, Daten schreiben/lesen, Datei schließen) analog den entsprechenden Basic-Befehlen arbeiten.

Das Besondere beim Datei-Handling auf Maschinenspracheebene ist, die Betriebssystemroutinen zu kennen, die zur Verwaltung logischer Dateien vorhanden sind. Das Ziel dieses Artikels ist es, Sie mit diesen Routinen vertraut zu machen. Da sich der Artikel an C 64-Besitzer wendet, die zum großen Teil über ein Diskettenlaufwerk verfügen, behandele ich die Datasette nur am Rande und bespreche auch die Behandlung relativer Dateien und die sogenannten »Direktzugriffs-Befehle«.

Zuvor will ich Ihnen jedoch noch erläutern, wieso es sehr vorteilhaft ist, die Dateiverwaltung in Maschinensprache zu beherrschen, selbst wenn Sie nicht vorhaben sollten, komplette Anwenderprogramme in Maschinensprache zu schreiben.

Die meisten größeren Programme für den C 64 werden Sie wahrscheinlich in Basic schreiben, da es außerordentlich aufwendig – und fehlerbehaftet – ist, ein größeres Programm vollständig in Maschinensprache zu erstellen. Am häufigsten wird Maschinensprache für kleinere Unterprogramme verwendet, um zeitkritische Basic-Programmteile zu ersetzen. Paradebeispiel für dieses Einsatzgebiet sind Sortierroutinen, die – in Basic programmiert – bei großen Datenmengen oftmals recht einschläfernd sein können.

Ein weiteres Einsatzgebiet sind Assemblerroutinen, die Mängel des eingebauten Basic beheben sollen, zum Beispiel eigene INPUT-Routinen, da der Basic-Befehl INPUT äußerst unzulänglich arbeitet (während der Eingabe kann der Bildschirm gelöscht werden; der Cursor kann aus dem Eingabefeld herausbewegt werden; es können maximal nur 80 Zeichen eingegeben werden und so weiter).

Viele nützliche Assemblerroutinen zum Einsatz in Basic-Programmen können jedoch nur dann erstellt werden, wenn Sie die Datenein- und -ausgabe auf der Maschinenspracheebene beherrschen. Vorstellbar wäre zum Beispiel ein sogenannter Drucker-Spooler, ein Assemblerprogramm, das in den Systeminterrupt eingebunden ist und das bei jedem Interrupt ein Zeichen eines Programmlistings oder Textes an den Drucker schickt. Während der Text durch diese »im Hintergrund laufende« Interruptroutine ausgedruckt wird, können Sie ungehindert mit Ihrem C 64 arbeiten, zum Beispiel ein Basic-Programm ablaufen lassen oder editieren.

Ein weiteres Beispiel ist eine INPUT#-Routine, die einen String von Kassette oder Diskette einliest, und die unter anderem in diesem Artikel vorgestellt werden wird. Wozu, werden Sie sich fragen, da doch das C 64-Basic den Befehl INPUT# bereits besitzt? Nun, jeder der sich bereits intensiver mit Dateiverwaltung beschäftigt hat, weiß, daß INPUT# den Programmierern des Basic-Interpreters gewiß keinen Ruhm einbrachte. Dieser Befehl arbeitet ebenso mangelhaft wie der Befehl INPUT: Soll ein String eingelesen werden, der länger ist als 80 Zeichen, erscheint die Fehlermeldung »STRING TOO LONG ERROR«. Einzulesende Strings dürfen weder ein Komma noch ein Semikolon enthalten, da beide Zeichen ebenso wie ein <RETURN> (CHR$(13)) als String-Endemarke aufgefaßt werden.

Im folgenden Artikel werde ich die wichtigsten Betriebssystemroutinen zur Datenein- beziehungsweise Datenausgabe vorstellen. Bei der Beschreibung dieser Routinen werden die Funktionen erläutert, wird erklärt, was bei Verwendung der jeweiligen Routine beachtet werden muß (Startadresse, Übergabeparameter), und welche Parameter – die vor allem zur Erkennung eventuell aufgetretener Fehler dienen – nach dem Aufruf der Routinen zurückübergeben werden.

Dieser Artikel setzt übrigens voraus, daß Sie sowohl über Maschinensprachekenntnisse als auch über Kenntnisse im Umgang mit sequentiellen und relativen Dateien verfügen. Wenn Sie bisher noch nichts vom »Positionieren«, von »Records« und »Kanälen« gehört haben, müssen Sie jedoch nicht gleich resignieren: In diesem Sonderheft finden Sie ab Seite 25 einen Artikel, der sich ausführlich mit den (Basic-) Grundlagen der verschiedenen Datei- und Zugriffsformen beschäftigt.

Zur Eingabe der Programmbeispiele: Die meisten der vorgestellten Routinen bestehen aus nur drei oder vier Maschinensprachebefehlen. Routinen, die etwas umfangreicher sind, wurden als Disassembler-Listings in den Text eingefügt und können mit jedem Monitor von Ihnen eingegeben werden.

Eine Ausnahme stellen die Programme »INPUT#-Routine« und »Verwaltung relativer Dateien« dar. Diese beiden Programme sind zu umfangreich, um als Disassembler-Listings noch problemlos verstanden zu werden. Sie wurden im Text als Assembler-Sourcecode (Hypra-Ass-Format) abgedruckt, da die Verwendung von Labeln in größeren Routinen doch einiges zur besseren Verständlichkeit beiträgt.

Die drei interessantesten Demoprogramme »INPUT#«, »Verwaltung relativer Dateien« und »ID ändern« sind zusätzlich als MSE-Listings vorhanden.

Öffnen einer Datei

Sie alle kennen den Basic-Befehl »OPEN«, mit dem eine Datei für Lese- oder Schreibzugriffe geöffnet werden kann. Erst nach dem Öffnen einer Datei kann die eigentliche Dateneinoder -ausgabe mit PRINT# oder INPUT# erfolgen. Die Programmierung auf der Maschinenspracheebene verläuft analog. Zuerst muß eine Datei geöffnet werden, wobei die gleichen Dateiparameter angegeben werden wie im OPEN-Befehl, dessen genaue Syntax bekanntlich lautet:

```
OPEN (LF), (GA), (SA), "DATEINAME, TYP, MODUS"
```

LF = Logische Filenummer
GA = Geräteadresse
SA = Sekundäradresse

Dieser für die Dateibehandlung grundlegende Basic-Befehl kann leider nur mit großem Aufwand in die entsprechenden Maschinenbefehle umgesetzt werden. Sollten Sie annehmen, daß hierzu der Aufruf einer einzigen Betriebssystemroutine ausreicht, muß ich Sie leider enttäuschen. Im Betriebssystem existiert zwar eine Routine mit dem Namen »OPEN«. Bevor jedoch diese Routine aufgerufen werden kann, müssen zwei Vorbereitungsroutinen aufgerufen werden, mit denen die benötigten Dateiparameter übergeben werden.

Der zuerst aufzurufenden Routine SETPAR ($FFBA) werden die Parameter »logische Filenummer«, »Geräteadresse« und »Sekundäradresse« übergeben. Anschließend wird die Routine SETNAM ($FFBD) aufgerufen, die für die Übergabe des Dateinamens zuständig ist (inklusive der Zusätze »Typ« und »Modus«). Erst nach dieser Vorbereitung kann die OPEN-Routine ($FFC0) aufgerufen werden.

SETPAR ($FFBA): Fileparameter setzen

Der erste Schritt zum Öffnen einer Datei – gleich ob zum Lesen oder Schreiben von Daten – ist die Übergabe der Parameter »log. Filenummer«, »Geräteadresse« und »Sekundäradresse« mit der Routine SETPAR.

Als log. Filenummer kann eine beliebige Zahl zwischen 1 und 255 angegeben werden.

Die Geräteadresse liegt üblicherweise zwischen Null und Acht (0=Tastatur; 1=Datasette; 3=Bildschirm; 4=Drucker; 8=Floppystation). In den folgenden Programmbeispielen werde ich immer (!) die Geräteadresse Acht verwenden, da die Floppystation wohl das am häufigsten in Verbindung mit dem C 64 verwendete Peripheriegerät ist.

Die Bedeutung der Sekundäradresse ist je nach verwendetem Peripheriegerät völlig unterschiedlich. Soll eine Diskettendatei geöffnet werden, kann eine beliebige Sekundäradresse zwischen 0 und 15 angegeben werden. Die Sekundäradresse 15 hat eine besondere Bedeutung, auf die ich noch genauer eingehen werde. Sie öffnet den sogenannten »Befehls-« oder auch »Fehlerkanal«.

Bei Druckern bestimmt die angegebene Sekundäradresse häufig den Druckmodus, ob zum Beispiel im Groß/Grafik- oder im Klein-/Großmodus gedruckt werden soll. Mit welcher Sekundäradresse welcher Druckmodus angesprochen wird, ist von Drucker zu Drucker unterschiedlich und muß daher im zugehörigen Druckerhandbuch nachgelesen werden.

Beim Öffnen einer Datasettendatei kann eine Sekundäradresse zwischen Null und Zwei angegeben werden. Null bedeutet, daß anschließend Daten vom Band gelesen werden sollen, eine Eins geben Sie an, wenn Schreibzugriffe erfolgen sollen. Die Angabe einer Zwei bedeutet ebenfalls, daß anschließend Schreibzugriffe erfolgen, zusätzlich wird jedoch eine sogenannte »EOT«-Markierung (= end of tape) auf das Band geschrieben, eine Markierung, die angibt, daß sich die Datei am Bandende befindet und keine weitere Datei folgt. Sinn und Zweck dieser Markierung blieb mir persönlich jedoch bislang etwas schleierhaft, da ich sie in meinen Programmen noch nie benötigt habe.

Sie wissen nun, welche Parameter der Routine SETPAR übergeben werden müssen, nicht jedoch wie. Alle drei Parameter werden mit Hilfe der Prozessorregister übergeben:

Akku = logische Filenummer (1-255)
X-Register = Geräteadresse (0, 1, 3, 4 oder 8)
Y-Register = Sekundäradresse (Datasette: 0-2; Floppy: 0-15)

Wenn wir zum Beispiel Daten in eine Diskettendatei schreiben wollen, übergeben wir der Routine SETPAR folgende Parameter:

```
LDA #$01  ;LOG.FILENUMMER 1
TAY       ;SEKUNDÄRADRESSE 1
LDX #$08  ;GERÄTEADRESSE 8 = FLOPPYSTATION
JSR $FFBA ;SETPAR AUFRUFEN
```

SETNAM ($FFBD): Dateinamen (+Zusätze) übergeben

Der Dateiname darf aus maximal 16 Zeichen bestehen. Unter »Zusatz« sind die Angaben »Typ« und »Modus« zu verstehen. Beide Angaben werden nur bei Diskettendateien benötigt, um zum Beispiel mit dem Befehl OPEN 2,8,2,"TEST,S,W" die sequentielle Datei »TEST« zum Schreiben zu öffnen.

Typ = S (Sequentielle Datei), R (Relative Datei), P (Programmdatei) oder U (Userdatei).

Modus = R (Read=Daten lesen), W (Write=Daten schreiben) oder A (Append=Daten an bestehende Datei anhängen)

Die Angabe des Dateityps wird bei Datasettendateien nicht benötigt, da die Datasette nur eine einzige Form der Datendatei kennt, den sequentiellen Typ.

Die Angabe der Zugriffsart, des »Modus«, ist bei Verwendung der Datasette ebenfalls überflüssig, da diese Angabe bereits beim Aufruf von SETPAR erfolgte (Sekundäradresse 0 bis 2).

Bei Verwendung der Datasette muß übrigens nicht unbedingt ein Dateiname angegeben werden. Die angelegte Datei erhält dann keinen Namen.

Die Parameter für SETNAM werden teilweise ebenfalls mit Hilfe der Prozessorregister übergeben. Zusätzlich muß jedoch in einem beliebigen Speicherbereich (im Programm selbst, in der Zeropage, oder an einem beliebigen anderen Ort) der Dateiname (plus Zusatz) in ASCII-Form abgelegt sein. In den folgenden Beispielen werde ich den Kassettenpuffer verwenden ($033C-$03FB), da dieser bei Verwendung der Floppystation nicht benötigt wird. Wenn Sie meine Demoprogramme für die Datasette umschreiben wollen, müssen Sie sich einen anderen Speicherbereich zur Datenablage suchen.

Wenn der Dateiname abgelegt wurde, wird der Akku mit der Länge des Namens geladen, und im X- beziehungsweise Y-Register wird ein Zeiger auf die Adresse übergeben, an der sich der Name befindet (X-Register = Low-Byte der Adresse; Y-Register = High-Byte):

```
LDA #$04 ;LÄNGE DES NAMENS: 4 ZEICHEN (Beispiel)
LDX #$3C ;LOW-BYTE DER ADRESSE $033C
LDY #$03 ;HIGH-BYTE DER ADRESSE $033C
JSR $FFBD ;SETNAM AUFRUFEN
```

Diese Routine ist nur dann lauffähig, wenn zuvor ein Dateiname mit einer Länge von vier Zeichen in ASCII-Form ab $033C abgelegt wurde.

OPEN ($FFC0): Datei öffnen

Nach diesen Vorbereitungen kann endlich die OPEN-Routine des Betriebssystems aufgerufen werden. Der Aufruf mit JSR $FFC0 genügt, da keine weiteren Parameter übergeben werden müssen. Die logische Datei wird nun geöffnet. Dabei ist es jedoch möglich, daß ein Fehler auftritt, zum Beispiel wenn auf Band geschrieben werden soll und die Datasette noch im Schrank liegt. In diesem Fall tritt ein »DEVICE NOT PRESENT ERROR« auf.

Wenn Sie wirklich professionell programmieren wollen, dürfen Sie eventuell auftretende Fehler selbstverständlich

nicht einfach ignorieren, sondern benötigen eine »Fehlerbehandlungsroutine«, zu der Ihr Programm in einem solchen Fall verzweigt (zum Beispiel, um den Benutzer aufzufordern, endlich die Diskettenstation einzuschalten oder eine Diskette einzulegen).

Ob beim Öffnen einer Datei ein Fehler autrat, erkennen Sie am Carry-Flag. Bei fehlerfreiem Öffnen der Datei ist es nach der Rückkehr aus der OPEN-Routine gelöscht. Trat ein Fehler auf, ist das Carry-Flag gesetzt und im Akku wird die Nummer des Fehlers übergeben. Bei gesetztem Carry-Flag sollte Ihr Programm daher zu einer Fehlerbehandlungsroutine verzweigen:

```
JSR $FFC0 ;OPEN AUFRUFEN
BCS $???? ;WENN FEHLER, DANN VERZWEIGEN
```

Die Fehlernummern im Akku haben folgende Bedeutung:

0 = <STOP>-Taste gedrückt (»Break Error«)
1 = too many files
2 = file open
3 = file not open
4 = file not found
5 = device not present
6 = not input file
7 = not output file
8 = missing filename
9 = illegal device number

Die Fehler Drei, Sechs und Sieben sind beim Öffnen einer Datei bedeutungslos. Sie können nur nach Lese- oder Schreibversuchen in die Datei auftreten. Ein Fehler wie zum Beispiel »MISSING FILENAME« ist dagegen gerätespezifisch. Bei Verwendung der Datasette müssen Sie nicht unbedingt einen Dateinamen angeben, daher kann auch dieser Fehler nicht auftreten.

CHKIN ($FFC6) und CKOUT ($FFC9): Ein-/Ausgaben umleiten

Ich muß gestehen, daß ich ein wenig vereinfachte mit der Behauptung, daß das Maschinensprache-Äquivalent zum Basic-Befehl OPEN die Routinen SETPAR, SETNAM und der anschließende Aufruf der OPEN-Routine sei. Leider müssen wir noch zwei weitere Routinen besprechen, bevor wir zur eigentlichen Datenein- beziehungsweise -ausgabe kommen, die Routinen CHKIN und CKOUT.

Wie Sie wissen, ist die Tastatur beim C 64 das Standardeingabegerät und der Bildschirm das Standardausgabegerät. Um nach dem Öffnen einer Datei in diese Daten zu schreiben oder Daten daraus zu lesen, muß die Ein- beziehungsweise Ausgabe auf die geöffnete logische Datei umgelenkt werden, da ansonsten die Standardgeräte angesprochen werden.

Das Umlenken der Eingabe von der Tastatur auf die mit der OPEN-Routine geöffnete Datei erfolgt mit der Routine CHKIN, der im X-Register die beim Aufruf von SETFLS verwendete logische Filenummer übergeben wird, zum Beispiel die Eins:

```
LDX #$01 ;LOG.FILENUMMER
JSR $FFC6 ;AUFRUF VON CHKIN
```

Um in eine geöffnete Datei Daten zu schreiben, muß die Ausgabe vom Standardgerät Bildschirm zuvor auf die geöffnete logische Datei umgeleitet werden. Der zuständigen Routine CKOUT wird ebenfalls im X-Register die verwendete logische Filenummer übergeben:

```
LDX #$01 ;LOG.FILENUMMER
JSR $FFC9 ;AUFRUF VON CKOUT
```

Beide Routinen zeigen auftretende Fehler ebenso wie die OPEN-Routine durch ein gesetztes Carry-Flag an. Trat ein Fehler auf, wird die Fehlernummer ebenfalls im Akku übergeben.

Demoprogramm 1: Datei in Maschinensprache öffnen

Wir sind nun endlich soweit, das theoretisch erworbene Wissen in ein Programm umsetzen zu können. Mit den erläuterten Routinen können wir ein Demonstrationsprogramm schreiben, das in Maschinensprache eine logische Datei zum Schreiben von Daten auf Diskette öffnet. Als Dateinamen verwenden wir »TEST,S,W« (das heißt der eigentliche Dateiname ist »TEST«, der Namenszusatz »,S,W«, da wir eine sequentielle Datei zum Schreiben öffnen wollen), als logische Filenummer eine Eins.

Zum Schreiben der Daten verwenden wir ein kleines Basic-Programm, das zusätzlich die Aufgabe übernehmen soll, den Dateinamen in ASCII-Form in den Kassettenpuffer zu POKEn, bevor es unsere Dateiöffnungsroutine aufruft und anschließend Daten in die – durch die Maschinenroutine geöffnete – Datei schreibt:

```
100 REM *** DATEN SCHREIBEN ***
110 DN$="TEST,S,W":REM DATEINAME + ZUSATZ
120 FOR I=1 TO LEN(DN$)
130 : POKE 827+I,ASC(MID$(DN$,I,1))
140 NEXT
150 SYS 49152:REM MASCHINENROUT. ($C000) AUFRUFEN
160 PRINT#1,"DIES IST"
170 PRINT#1,"EIN TEST"
180 CLOSE 1
190 :
200 REM *** DATEN LESEN ***
210 OPEN 1,8,2,"TEST,S,R":REM DATEI OEFFNEN
220 INPUT#1,A$:PRINT A$
230 INPUT#1,B$:PRINT B$
240 CLOSE 1
```

Die Maschinenroutine beginnt ab $C000, ebenso wie alle weiteren Demoprogramme.

```
. C000  LDA #$01   ;LOG.FILENUMMER 1
. C002  LDX #$08   ;GERÄTEADRESSE 8 = FLOPPY
. C004  LDY #$02   ;SEKUNDÄRADRESSE 2
. C006  JSR $FFBA  ;SETPAR AUFRUFEN

. C009  LDA #$08   ;LÄNGE DES NAMENS+ZUSATZ:
                    8 ZEICHEN
. C00B  LDX #$3C   ;LOW-BYTE DER ADRESSE $033C
. C00D  LDY #$03   ;HIGH-BYTE DER ADRESSE $033C
. C00F  JSR $FFBD  ;SETNAM AUFRUFEN

. C012  JSR $FFC0  ;OPEN AUFRUFEN

. C015  LDX #$01   ;LOG.FILENUMMER
. C017  JSR $FFC9  ;AUFRUF VON CKOUT
. C01A  RTS        ;ZURUECK NACH BASIC
```

Das Maschinenprogramm besteht im Grunde nur aus einer Zusammenfassung der bereits beschriebenen Teile, das heißt aus dem Aufruf von SETPAR, SETNAM, OPEN und CKOUT. Beachten Sie bitte, daß eventuell auftretende Fehler ignoriert werden (Carry-Flag nach OPEN und CKOUT abfragen).

Geben Sie dieses Programm bitte mit einem Monitor ein (lassen Sie die Kommentare jedoch weg und versuchen Sie nicht, die einzelnen Programmteile wie abgebildet durch Leerzeilen optisch zu unterteilen!), geben Sie anschließend das Basic-Programm ein und starten Sie es mit RUN.

Zum Programmablauf: In den Zeilen 110-140 werden mit einer Schleife die ASCII-Codes der einzelnen Zeichen des Dateinamens »TEST« und des Zusatzes »,S,W« in den Kassettenpuffer gePOKEt. Anschließend wird unsere Maschinenroutine aufgerufen, die eine Diskettendatei unter der log. Filenummer Eins öffnet (Zeile 150). Unter Angabe dieser File-

nummer schreibt das Basic-Programm nun zwei Strings in die Datei, bevor sie wieder geschlossen wird (Zeilen 160 bis 180).

Anschließend wird die sequentielle Datei »TEST« zum Lesen geöffnet (Zusatz »,S,R«). Die beiden Strings werden eingelesen, auf dem Bildschirm ausgegeben und die Datei geschlossen.

Wie Sie sehen, ist es zweifellos bequemer, einen Basic-Befehl zum Öffnen einer Datei zu verwenden, als ein Maschinenprogramm, das an mehrere Routinen die verschiedensten Parameter übergeben muß, um die gleiche Funktion zu erfüllen. Die im Laufe dieses Kapitels vorgestellte INPUT#-Routine entschädigt Sie jedoch für diesen Aufwand. In Basic könnte diese Routine nur mit einem kleinen Unterprogramm unter Verwendung von GET# erstellt werden und wäre weitaus langsamer als die entsprechende Maschinenroutine. (Versuchen Sie einmal, 20 Strings mit einer Länge von jeweils 100 Zeichen mit GET# nach und nach einzulesen. Sie werden staunen, wie langsam Ihre Datasette plötzlich wird.)

BSOUT ($FFD2): Datenausgabe auf logische Datei

Wir können nun zwar in Maschinensprache beliebige Dateien öffnen, bisher jedoch weder Daten ausgeben noch einlesen. Sie alle wissen sicherlich, daß mit der Routine BSOUT Daten auf dem Bildschirm ausgegeben werden können. BSOUT gibt ein im Akku übergebenes Zeichen (ASCII-Code!) auf dem Bildschirm aus, und zwar an der aktuellen Cursorposition.

BSOUT ist jedoch keineswegs eine speziell zur Ausgabe auf dem Bildschirm gedachte Routine, sondern erfüllt die weitaus allgemeinere Funktion, Daten in eine beliebige logische Datei zu schreiben. Diese Datei ist normalerweise der Bildschirm, wir können mit BSOUT jedoch auch Zeichen in eine Datasetten-, Disketten- oder Druckerdatei schreiben.

Die benötigten Vorbereitungsroutinen kennen wir bereits. Zuerst wird eine logische Datei geöffnet, anschließend die Datenausgabe mit CKOUT auf diese Datei umgeleitet. Wenn nun BSOUT aufgerufen wird, werden die im Akku übergebenen Zeichen dank der Umleitung der Ausgabe nicht auf dem Bildschirm ausgegeben, sondern in die angegebene Datei geschrieben.

```
...             ;DISKETTENDATEI ZUM
...             ;SCHREIBEN OEFFNEN
JSR $FFC9       ;AUSGABE AUF DIE DATEI LEGEN
LDA #$41        ;ASCII-CODE VON 'A'
JSR $FFD2       ;'A' IN DIE DATEI SCHREIBEN
...
...
...
```

BASIN ($FFCF): Dateneingabe von logischer Datei

BASIN erfüllt die entgegengesetzte Funktion, das Einlesen von Daten aus einer beliebigen logischen Datei. Wenn die Zeichen nicht vom Standardgerät Tastatur gelesen werden sollen, muß ebenfalls eine logische Datei zum Lesen geöffnet und die Eingabe mit CHKIN auf diese Datei umgelenkt werden. Nun kann mit BASIN Zeichen für Zeichen aus dieser Datei gelesen werden. Die eingelesenen Zeichen werden im Akku übergeben.

```
...             ;DISKETTENDATEI ZUM
...             ;LESEN OEFFNEN
JSR $FFC6       ;EINGABE AUF DIE DATEI LEGEN
JSR $FFCF       ;EIN ZEICHEN AUS DER DATEI LESEN
STA $xxx1       ;ZEICHEN SPEICHERN
JSR $FFCF       ;NAECHSTES ZEICHEN LESEN
STA $xxx2       ;EBENFALLS SPEICHERN
...
...
```

CLRCH ($FFCC): Standardein-/-ausgabegeräte setzen

Nach beendeter Ein- beziehungsweise Ausgabe von Daten sollte die Routine CLRCH aufgerufen werden, die die Eingabe wieder auf das Standardgerät Tastatur und die Ausgabe auf den Bildschirm umleitet. Es werden keinerlei Vorbereitungsroutinen oder Übergabeparameter benötigt. CLRCH muß auch aufgerufen werden, wenn Ein-/Ausgabeoperationen noch nicht beendet wurden, sich im folgenden jedoch auf andere logische Dateien beziehen sollen, das heißt bevor die Ein- beziehungsweise Ausgabe mit CKOUT oder CHKIN auf eine andere Datei umgelenkt wird. Ohne die vorhergehende »Säuberung« der Übertragungskanäle durch CLRCH kann es passieren, daß mehrere Geräte gleichzeitig auf den seriellen Bus zugreifen wollen, was verständlicherweise zu erheblichen Problemen führt.

```
...             ;LOGISCHE DATEI OEFFNEN
...             ;EIN- ODER AUSGABE UMLENKEN
...             ;DATEN LESEN ODER SCHREIBEN
JSR $FFCC       ;STANDARDEIN-/AUSGABEGERAETE SETZEN
...             ;(TASTATUR/BILDSCHIRM)
...
...
```

CLOSE ($FFC3): Logische Datei schließen

CLRCH leitet weitere Ein- oder Ausgaben zwar wieder auf die Tastatur beziehungsweise den Bildschirm um, die zuvor geöffnete logische Datei wird jedoch nicht automatisch geschlossen! Zum Abschluß jeder Schreib- oder Leseroutine sollte daher die Routine CLOSE aufgerufen werden, die die angegebene logische Datei schließt, wobei die Filenummer im Akku übergeben werden muß.

```
...             ;DATEI MIT LOG.FILENUMMER 1 OEFFNEN
...             ;EIN-/AUSGABE UMLEITEN
...             ;DATEN LESEN ODER SCHREIBEN
JSR $FFCC       ;EIN-/AUSGABE AUF STANDARDGERAETE
LDA #$01        ;LOG.FILENUMMER
JSR $FFC3       ;LOG.DATEI SCHLIESSEN
...
...
...
```

READST ($FFB7): Status abfragen

Ein Problem müssen wir noch lösen, bevor wir ausschließlich in Maschinensprache Dateiverwaltung betreiben können, und zwar die Abfrage des Dateiendes. Das Einlesen einer Datei muß beendet werden, wenn das Dateiende erreicht wurde. In Basic können wir hierzu die Statusvariable ST abfragen, die den Wert 64 annimmt, wenn das Dateiende erreicht wurde.

In Maschinensprache verwenden wir die Routine READST, die die gleiche Funktion erfüllt (der Basic-Interpreter verwendet diese Routine selbst, um der Variablen ST den jeweiligen Status mitzuteilen).

Nach dem Aufruf von READST wird der aktuelle Status im Akku übergeben. Bei Erreichen des Dateiendes wird Bit 6 gesetzt (da dieses Bit den dezimalen Wert 64 besitzt, wird die Analogie zur Basic-Variablen ST deutlich).

```
...
...
...
LABEL JSR $FFCF  ;DATEN LESEN
      JSR $FFB7  ;READST AUFRUFEN
      AND #$40   ;ALLE BITS AUSSER
                  BIT 6 AUSMASKIEREN
      BEQ LABEL  ;WEITERLESEN, WENN NICHT DATEIENDE
...
...
...
```

Demoprogramm 2: Daten in Maschinensprache schreiben und lesen

Die Zeiten, in denen wir uns zum Schreiben und Lesen von Daten mit Basic behelfen mußten, sind nun endgültig vorbei. In diesem zweiten Programmbeispiel werden wir die Strings »DIES IST« und »EIN TEST« in Maschinensprache auf Band schreiben und wieder lesen.

Aus Bequemlichkeit wird zusätzlich ein kleines Basic-Programm verwendet, das den Dateinamen »TEST« und den Zusatz »,S,W« nach $033C-$0343 (dezimal 828-835) und die zu schreibenden Strings ab $034C (dezimal 828+16) in den Kassettenpuffer POKEt (Zeilen 110 bis 130), und das die Routine zum Schreiben der Strings aufruft (Zeile 160). Vor dem Aufruf der Leseroutine POKEt das Basic-Programm den gleichen Dateinamen »TEST«, jedoch mit dem geänderten Zusatz »,S,R« nach $033C, da die Datei nun zum Lesen geöffnet werden soll.

```
100 REM *** STRINGS VORBEREITEN ***
110 FOR I=1 TO 8:POKE 827+I,ASC(MID$("TEST,S,W",I,1
    )):NEXT
120 A$="DIES IST"+CHR$(13)+"EIN TEST"+CHR$(13)+CHR
    $(0)
130 FOR I=1 TO LEN(A$):POKE 827+16+I,ASC(MID$(A$,I
    ,1)):NEXT
140 :
150 REM *** STRINGS SCHREIBEN/LESEN ***
160 SYS 49152:REM STRINGS SCHREIBEN
170 FOR I=1 TO 8:POKE 827+I,ASC(MID$("TEST,S,R",I,
    1)):NEXT
180 SYS 49200:REM STRINGS LESEN/AUSGEBEN
```

Beachten Sie bitte, daß bei der zeichenweisen Ausgabe mit BSOUT im Gegensatz zur Basic-Ausgabe mit PRINT# nicht automatisch nach jedem String das Zeichen CHR$(13) (Carriage Return oder auch Zeilenvorschub) auf Kassette geschrieben wird. Die Basic-Routine fügt dieses Zeichen daher an das Ende beider Zeichenketten an (Zeile 120). Als letztes Zeichen wird ein CHR$(0) als Endemarke angehängt, an dem unsere Maschinenroutine das Ende des auszugebenden Strings erkennen soll.

Teil 1: Daten schreiben

```
. C000  LDA #$01     ;LOG.FILENUMMER
. C002  LDX #$08     ;GERAETEADRESSE
. C004  LDY #$02     ;SEKUNDAERADRESSE
. C006  JSR $FFBA    ;SETPAR AUFRUFEN

. C009  LDA #$08     ;LAENGE DES DATEINAMENS+
                      ZUSATZ
. C00B  LDX #$3C     ;LOW-BYTE ADRESSE DATEINAME
. C00D  LDY #$03     ;HIGH-BYTE ADRESSE DATEINAME
. C00F  JSR $FFBD    ;SETNAM AUFRUFEN
. C012  JSR $FFC0    ;OPEN AUFRUFEN
. C015  LDX #$01     ;LOG.FILENUMMER
. C017  JSR $FFC9    ;CKOUT AUFRUFEN

. C01A  LDX #$00     ;ZEIGER INITIALISIEREN
. C01C  LDA $034C,X  ;ERSTES ZEICHEN HOLEN
. C01F  BEQ $C027    ;FERTIG, WENN ENDEMARKE
. C021  JSR $FFD2    ;ZEICHEN IN DATEI SCHREIBEN
. C024  INX          ;ZEIGER AUF NAECHSTES ZEICHEN
. C025  BNE $C01C    ;ZUM SCHLEIFENANFANG
                      (UNBEDINGTER SPRUNG)

. C027  JSR $FFCC    ;STANDARDGERAETE SETZEN
. C02A  LDA #$01     ;LOG.FILENUMMER
. C02C  JSR $FFC3    ;LOG.DATEI SCHLIESSEN
. C02F  RTS          ;RUECKKEHR NACH BASIC
```

Teil 2: Daten lesen/ausgeben

```
. C030  LDA #$01     ;LOG.FILENUMMER
. C032  LDX #$08     ;GERAETEADRESSE
. C034  LDY #$02     ;SEKUNDAERADRESSE
. C036  JSR $FFBA    ;SETPAR AUFRUFEN

. C039  LDA #$08     ;LAENGE DES DATEINAMENS
. C03B  LDX #$3C     ;LOW-BYTE ADRESSE DATEINAME
. C03D  LDY #$03     ;HIGH-BYTE ADRESSE DATEINAME
. C03F  JSR $FFBD    ;SETNAM AUFRUFEN

. C042  JSR $FFC0    ;OPEN AUFRUFEN
. C045  LDX #$01     ;LOG.FILENUMMER
. C047  JSR $FFC6    ;CHKIN AUFRUFEN

. C04A  JSR $FFCF    ;BASIN AUFRUFEN
. C04D  JSR $FFD2    ;ZEICHEN AUF BILDSCHIRM
                      AUSGEBEN
. C050  JSR $FFB7    ;READST AUFRUFEN
. C053  AND #$40     ;ALLE AUSSER BIT 6
                      AUSMASKIEREN
. C055  BEQ $C04A    ;WEITER, WENN NICHT DATEIENDE

. C057  JSR $FFCC    ;STANDARDGERAETE SETZEN
. C05A  LDA #$01     ;LOG.FILENUMMER
. C05C  JSR $FFC3    ;LOG.DATEI SCHLIESSEN
. C05F  RTS          ;RUECKKEHR NACH BASIC
```

Dieses Programm dürfte gut verständlich sein, da es unter Verzicht auf Programmiertricks in aller Ausführlichkeit geschrieben wurde. Mit dem Öffnen und Schließen der Dateien im ersten und zweiten Programmteil sollten Sie nun vertraut sein.

Wichtig an diesem Programm ist, daß im ersten Teil Zeichen für Zeichen in die Datei geschrieben wird, bis das nächste Zeichen eine Null ist, jene Endemarke, die das Basic-Programm am Ende des Strings in den Speicher POKEte.

Beim Lesen der Zeichen könnte selbstverständlich ebenfalls eine solche Endemarke benutzt werden, um das Dateiende festzustellen. Statt dessen wird die erläuterte Routine READST verwendet. Zeichen für Zeichen wird aus der Datei gelesen und auf dem Bildschirm ausgegeben, wobei nach jedem Lesevorgang READST aufgerufen und das sechste Bit des im Akku übergebenen Statusbytes überprüft wird (AND #$40). Ist dieses Bit gesetzt, wurde das Dateiende erreicht und das Programm kehrt ins Basic zurück (zuvor werden noch CLRCH und CLOSE aufgerufen).

Bevor Sie dieses Programm starten: Sollten Sie das Demoprogramm 1 ausprobiert haben, so löschen Sie bitte die dadurch auf der Diskette erzeugte Datei »TEST«.

Verwenden Sie keinesfalls den Zusatz »@:« im Dateinamen, um bereits existierende Dateien zu überschreiben, da er unter Umständen Ihre Diskette zerstören kann.

Die Interpreterroutinen CHKKOM, GETBYT, STRPOS und STRRES

Zu Beginn dieses Artikels erwähnte ich den Basic-Befehl INPUT#, der als »Fehlleistung« des Basic-Interpreters angesehen werden kann. Im folgenden wird eine Routine entwickelt, die diesen Befehl durch eine Maschinenroutine ersetzt und die 255 beliebige Zeichen einlesen kann (also auch Komma, Doppelpunkt etc.).

Vorher muß ich jedoch kurz auf verschiedene Routinen des Basic-Interpreters eingehen, die für diese Routine unbedingt benötigt werden. Da der Aufruf möglichst weitgehend dem normalen INPUT#-Befehl entsprechen soll (INPUT#(log. Filenummer), (Übergabestring)), muß das Programm in der Lage sein, die Filenummer aus dem Basic-Text zu lesen und auf den angegebenen String zuzugreifen, in dem die eingelesenen Daten an das aufrufende Basic-Programm übergeben werden sollen.

Die Routine wird ebenfalls im Bereich $C000 liegen und wie folgt aufgerufen:
SYS 49152,(LOG.FILENUMMER),(ÜBERGABESTRING)
zum Beispiel:
SYS 49152,1,a$ oder SYS 49152,2,X$(7)

Da die einzelnen Parameter durch Kommata getrennt sind, benötigen wir zuerst eine Routine, die in der Lage ist, Kommas aus dem Basic-Programm einzulesen. Diese Routine wird CHKKOM ($AEFD) genannt. Der Basic-Interpreter führt ständig einen Zeiger auf die momentan bearbeitete Textstelle mit sich. Nach dem SYS-Aufruf weist dieser Zeiger auf das erste Zeichen hinter der angegebenen Adresse, das heißt auf das Komma.

Der Aufruf von CHKKOM liest dieses Zeichen ein, setzt den Textpointer auf das nächste Zeichen im Basic-Programm und überprüft, ob tatsächlich ein Komma gelesen wurde. Wenn ja, endet CHKKOM mit einem RTS-Befehl, ansonsten wird ein »SYNTAX ERROR IN ...« ausgegeben.

Wenn mit CHKKOM das erste Trennzeichen eingelesen wurde, muß anschließend die angegebene Filenummer von unserem Programm gelesen werden. Die Routine GETBYT ($B79E) liest einen beliebigen Ein-Byte-Wert aus dem Basic-Text und übergibt ihn im X-Register. Der Wert muß übrigens nicht direkt als Zahl angegeben werden, da GETBYT auch Variablen verarbeitet, so daß zum Beispiel folgender Aufruf möglich ist:
LF=1:SYS 49152,LF,A$

Bevor wir an die eigentliche Programmerstellung gehen können, muß ein weiteres Problem gelöst werden: Wo speichern wir die eingelesenen Zeichen ab? Am komfortabelsten ist die Benutzung unserer Routine, wenn sie die eingelesenen Zeichen als String anlegt, auf den das aufrufende Basic-Programm direkt zugreifen kann. Der Name des Übergabestrings sollte ebenso wie beim INPUT#-Befehl beim Aufruf angegeben werden können.

Ein Standardproblem bei der Verbindung von Basic und Maschinenroutinen ist der Zugriff auf Basic-Strings von Maschinensprache aus, das Lesen oder gar Anlegen von Strings. Beides ermöglicht uns die Routine STRPOS ($B08B). Diese Routine liest den beim Aufruf angegebenen Stringnamen aus dem Basic-Text und übergibt in $47/$48 einen Pointer auf die sogenannten »Stringdescriptoren«.

Wie Sie vielleicht wissen, werden Strings vom Ende des verfügbaren Speichers aus abwärts angelegt und befinden sich im Gegensatz zu numerischen Variablen nicht direkt in der sogenannten »Variablentabelle«, die im Speicher immer unmittelbar dem Basic-Programm folgt. In der Variablentabelle befinden sich jedoch alle Daten, die zum Zugriff auf den eigentlichen String notwendig sind, der Stringname, die Stringlänge und die Adresse, an der sich der String selbst befindet (Name: 2 Byte; Länge: 1 Byte; Adresse: 2 Byte in der Form Low/High).

STRPOS übergibt einen Zeiger auf diese Descriptoren des angegebenen Strings (und zwar nicht auf das erste Descriptorenbyte (das erste Namensbyte), sondern auf das Längenbyte) und legt ihn an, wenn er bisher noch nicht existiert. Wenn der Aufruf zum Beispiel lautet: SYS 49152,A$(7) und unsere Routine zuerst CHKKOM und anschließend STRPOS aufruft, erhalten wir in $47/$48 einen Pointer, der auf den Längendescriptor des Strings A$(7) zeigt.

Die letzte benötigte Routine heißt STRRES ($B4F4). Diese Routine reserviert Speicherplatz für den anzulegenden String. Ein Beispiel: Wir wollen einen String mit einer Länge von 10 Zeichen anlegen. Zuerst muß der Akku mit der Stringlänge geladen werden, anschließend wird STRRES aufgerufen:

```
LDA #$0A ;STRINGLAENGE: 10 ZEICHEN
JSR $B4F4 ;AUFRUF VON STRRES
```

In $33/$34 befindet sich ein Pointer auf den Anfang des sogenannten »Stringstacks«, das heißt auf das erste Zeichen des zuletzt angelegten Strings. Nach dem Aufruf von STRRES mit der Stringlänge zehn im Akku wird dieser Pointer um den Wert zehn erniedrigt. Der dazukommende String kann nun vor (!) dem zuletzt angelegten gespeichert werden. (Denken Sie bitte daran, daß die Strings vom Speicherende ausgehend abwärts (!) angelegt werden.)

Bevor STRRES den Pointer auf den Anfang der Strings um die übergebene Stringlänge vermindert, wird jedoch überprüft, ob überhaupt ausreichend Platz vorhanden ist, oder ob der nach unten wachsende Stringstack mit dem Basic-Programm kollidieren und dieses überschreiben würde. Wenn nicht ausreichend Platz vorhanden ist, wird eine Garbage Collection durchgeführt, das heißt nicht mehr benötigter »Stringmüll« wird beseitigt. Ist auch anschließend noch nicht ausreichend Platz für den anzulegenden String vorhanden, gibt STRRES die Fehlermeldung »OUT OF MEMORY ERROR« aus.

Endlich: Die INPUT#-Routine

Nach diesem »Vorgeplänkel« können wir uns endlich an die eigentliche Programmierung wagen. Im Gegensatz zu den bisher verwendeten Monitorlistings werde ich die Entwicklung der INPUT#-Routine anhand von Assemblerlistings (Hypra-Ass-Format) demonstrieren, da diese Routine doch etwas komplexer als die bisherigen Programmbeispiele ist und Assemblerlistings einfacher zu durchschauen sind.

```
;*************************
;*    INPUT#FUER C64     *
;*  (W) SAID BALOUI, 1986 *
;*************************
.BA $C000              ;PROGRAMMSTART
;
.EQ LENGTH   = $F7     ;STRINGLAENGE
.EQ POINTR   = $F8     ;HILFSPOINTER
.EQ DESCR    = $FA     ;STRINGDESCRIPTOREN
.EQ DESPOI   = $47     ;POINTER AUF DESCRIPTOREN
.EQ CHKKOM   = $AEFD   ;KOMMA EINLESEN
.EQ GETBYT   = $B79E   ;BYTEWERT LESEN
.EQ CHKIN    = $FFC6   ;INPUT AUF LOGISCHE DATEI
.EQ CLRCH    = $FFCC   ;STANDARDGERAETE SETZEN
.EQ BASIN    = $FFCF   ;ZEICHEN AUS LOG.DATEI
                        LESEN
.EQ STREND   = $33     ;POINTER AUF STRINGSTACK
.EQ STRPOS   = $B08B   ;STRINGVARIABLE LESEN
.EQ STRRES   = $B4F4   ;STRINGPLATZ RESERVIEREN
;
;
```

```
*** PLATZ RESERVIEREN ***
            LDA #255      ;255 BYTE
            JSR STRRES    ;RESERVIEREN
```

Diese beiden Befehle reservieren Platz für den einzulesenden String in der Maximallänge von 255 Zeichen. Dank der Verwendung von STRRES wird sich das Programm keinesfalls »aufhängen«, sondern, sollte das Basic-Programm, in dem Sie die INPUT#-Routine verwenden, zu umfangreich und daher nicht ausreichend Platz vorhanden sein, sich mit einem »OUT OF MEMORY ERROR« verabschieden.

```
*** STRING EINLESEN ***
            JSR CHKKOM       ;LOG.FILENUMMER
            JSR GETBYT       ;...EINLESEN (X-REGISTER)
            JSR CHKIN        ;EINGABE AUF LOG.DATEI
            LDY #0
INPUT       JSR BASIN
            CMP #13          ;LESEN BIS »RETURN«
            BEQ LIESEND      ;UND AB STREND/STREND+1
            STA (STREND),Y   ;IM STRINGSTACK ABLEGEN
            INY
            BNE INPUT        ;UNBEDINGTER SPRUNG (!)
LIESEND     STY LENGTH       ;STRINGLAENGE MERKEN
            JSR CLRCH        ;STANDARDGERAETE SETZEN
```

Der Programmteil zum Einlesen des Strings liest zuerst das als Trennzeichen verwendete Komma und die angegebene Filenummer aus dem Basic-Text ein. GETBYT übergibt wie erwähnt die Filenummer im X-Register, in dem sich die Filenummer auch beim Aufruf von CHKIN befinden muß, so daß diese Routine ohne weitere Vorbereitung aufgerufen und die Eingabe auf die geöffnete Datei umgelenkt werden kann.

Das Y-Register wird nun mit Null initialisiert, Zeichen für Zeichen aus der Datei eingelesen und ab dem neuen Anfang des Stringstacks gespeichert, bis das Zeichen für <RETURN> gelesen wird, das das Stringende kennzeichnet. Die im Y-Register enthaltene Stringlänge wird in LENGTH zur späteren Verwendung gemerkt und durch Aufruf von CLRCH wieder die Standardgeräte (Tastatur, Bildschirm) gesetzt.

```
*** POINTER AUF STRINGANFANG ***
            LDA STREND+1     ;POINTER(+1)=
            STA POINTR+1     ;ECHTER STRINGANFANG,
            LDA #255         ;D.H. MOMENTANER
            SEC              ;ANFANG (STREND)
            SBC LENGTH       ;+ DIE DIFFERENZ
            CLC              ;ZWISCHEN DER
            ADC STREND       ;RESERVIERTEN LAENGE
            STA POINTR       ;255 ZEICHEN UND
            BCC NOINC        ;DER TATSAECHLICHEN
            INC POINTR+1     ;LAENGE DES STRINGS
```

Die eingelesenen Zeichen wurden ab der Adresse gespeichert, auf die STREND weist. STREND weist durch die Stringreservierung 255 Zeichen vor den Beginn des nächsten Strings im Stringstack. Um die unglaubliche Platzverschwendung zu vermeiden, wenn zwar 255 Zeichen reserviert wurden, der eingelesene String tatsächlich jedoch zum Beispiel nur eine Länge von 20 oder 30 Zeichen besitzt, wird nach beendetem Einlesen der String nach oben verschoben, um den Leerraum zwischen dem eingelesenen und dem nächsten String im Stringstack wiederzugewinnen. Dazu wird zuerst anhand der tatsächlichen Stringlänge LENGTH und der daraus resultierenden Differenz zum reservierten Speicherplatz ein Pointer POINTR errechnet, der auf die korrekte Adresse weist, an die der String verschoben werden muß.

```
;*** STRING VERSCHIEBEN ***
NOINC       LDY LENGTH       ;STRING VON:
            DEY              ;STREND BIS STREND+
                              LENGTH
COPY        LDA (STREND),Y   ;NACH:
            STA (POINTR),Y   ;POINTR BIS POINTR+
                              LENGTH
            DEY              ;KOPIEREN
            CPY #255
            BNE COPY
```

Das zeichenweise Kopieren des Strings dürfte problemlos zu verstehen sein. Der String befindet sich an der korrekten Adresse und unsere Routine wäre beendet. Doch leider kann das Basic-Programm noch nicht auf den angelegten String zugreifen, bevor nicht mehrere Pointer korrigiert werden.

```
;*** DESCRIPT/STREND BEHANDELN ***
            JSR CHKKOM
            JSR STRPOS
```

Mit CHKKOM wird das der Filenummer folgende Komma gelesen und STRPOS übergibt in DESPOI ($47/$48) einen Pointer auf den beim Aufruf angegebenen String. Die Descriptoren dieses Strings werden in einer Schleife nach DESCR (Länge), DESCR+1 (Adresse Low) und DESCR+2 (Adresse high) kopiert.

```
;*** DESCRIPTOREN/STREND BEHANDELN ***
            LDY #0           ;DESCRIPTOREN
                              AKTUALISIEREN
            LDA LENGTH       ;(LAENGENDESCRIPTOR
                              MIT DER
            STA (DESPOI),Y   ;STRINGLAENGE UND
                              ADRESSEN-
            INY              ;DESCRIPTOREN MIT
                              DER STRING-
            LDA POINTR       ;ADRESSE VERSEHEN
            STA (DESPOI),Y   ;AUSSERDEM STREND/
                              STREND+1
            STA STREND       ;ENTSPRECHEND DER
                              GEAENDERTEN
            INY              ;STRINGADRESSE
                              KORRIGIEREN
            LDA POINTR+1
            STA (DESPOI),Y
            STA STREND+1
            RTS              ;ZURUECK NACH BASIC
.EN                          ;PROGRAMMENDE
```

Der Längendescriptor des Strings wird nun mit der Länge LENGTH versehen und die beiden Adressendescriptoren (Low/High) mit der geänderten Startadresse POINTR/POINTR+1. Da der von den reservierten 255 Zeichen unbenutzte Bereich durch das Verschieben des Strings wieder freigegeben wurde, muß natürlich auch STREND/STREND+1 entsprechend der endgültigen Stringadresse korrigiert werden.

Alle Pointer und Descriptoren sind nun völlig korrekt gesetzt und das Basic-Programm kann auf den angelegten String zugreifen, wie das folgende Demoprogramm zeigt:

```
100 REM *** STRING ERZEUGEN ***
110 X$="123456789,"
120 FOR I=1 TO 25:Y$=Y$+X$:NEXT
130 Y$=Y$+"12345"
140 :
150 REM *** STRING SCHREIBEN ***
160 OPEN 1,8,2"TEST,S,W"
170 PRINT#1,Y$
180 CLOSE 1
190 :
200 REM *** STRING LESEN ***
210 OPEN 1,8,2,"TEST,S,R"
220 SYS 49152,1,A$
230 CLOSE 1
240 PRINT A$
```

Nach dem Starten mit RUN schreibt dieses Demoprogramm einen String in der Maximallänge von 255 Zeichen inklusive dem mit INPUT# keinesfalls einlesbaren Zeichen »,« auf Diskette und liest ihn mit der vorgestellten Routine in der vollen Länge mit Kommatas wieder ein, was für den

INPUT #-Befehl völlig unmöglich wäre. Das Basic-Programm kann auf den eingelesenen String wie auf jeden anderen String auch zugreifen, zum Beispiel mit »PRINT A$«.

Verwenden Sie zur Eingabe der INPUT #-Routine bitte das abgebildete MSE-Listing (Listing 1).

```
Name : c64.input#              c000 c05c
----------------------------------------
c000 : a9 ff 20 f4 b4 20 fd ae   f1
c008 : 20 9e b7 20 c6 ff a0 00   58
c010 : 20 cf ff c9 0d f0 05 91   e0
c018 : 33 c8 d0 f4 84 f7 20 cc   a4
c020 : ff a5 34 85 f9 a9 ff 38   0d
c028 : e5 f7 18 65 33 85 f8 90   20
c030 : 02 e6 f9 a4 f7 88 b1 33   a9
c038 : 91 f8 88 c0 ff d0 f7 20   26
c040 : fd ae 20 8b b0 a0 00 a5   69
c048 : f7 91 47 c8 a5 f8 91 47   ea
c050 : 85 33 c8 a5 f9 91 47 85   aa
c058 : 34 60 01 08 00 00 00 00   fe
```

Listing 1. Ein INPUT #-Befehl in Maschinensprache

Umgehung der Routinen SETPAR und SETNAM

Alle grundlegenden Routinen zur Arbeit mit logischen Dateien sind Ihnen nun bekannt. Bevor ich mich im folgenden verschiedenen speziellen Anwendungen zuwende (Fehlerkanal abfragen, relative Dateien etc.) will ich der Vollständigkeit wegen noch erläutern, wie eine logische Datei geöffnet werden kann, ohne (!) zuvor SETPAR und SETNAM aufzurufen.

Wenn die Register mit den jeweiligen Parametern geladen und SETPAR aufgerufen wird, kopiert diese Routine anschließend die übergebenen Werte in die Zeropageadressen $B8 (log. Filenummer), $BA (Geräteadresse) und $B9 (Sekundäradresse). Diese Parameter können Sie auch direkt in die angegebenen Speicherstellen schreiben.

Ebenso können Sie SETNAM umgehen, wenn Sie die Länge des Filenamens in $B7 und die Adresse in $BB/$BC ablegen. Nach diesen Vorbereitungen kann wie gewohnt die OPEN-Routine aufgerufen werden.

Mit diesen Kenntnissen kann das Demoprogramm 1 problemlos umgeschrieben werden:

```
. C000  LDA #$01     ;LOG.FILENUMMER
. C002  STA $B8      ;NACH $B8
. C004  LDA #$08     ;GERAETEADRESSE
. C006  STA $BA      ;NACH $BA
. C008  LDA #$02     ;SEKUNDAERADRESSE
. C00A  STA $B9      ;NACH $B9

. C00C  LDA #$08     ;LAENGE DES FILENAMENS
. C00E  STA $B7      ;NACH $B7
. C010  LDA #$3C     ;ADRESSE DES FILENAMENS:
. C012  LDX #$03     ;LOW-BYTE UND HIGH-BYTE
. C014  STA $BB      ;NACH $BB
. C016  STX $BC      ;BZW. $BC

. C018  JSR $FFC0    ;OPEN AUFRUFEN
. C01B  LDX #$01     ;LOG.FILENUMMER
. C01D  JSR $FFC9    ;CKOUT AUFRUFEN
. C020  RTS          ;ZURUECK NACH BASIC
```

Wenn Sie die Routinen SETPAR und SETNAM aus irgendwelchen Gründen nicht verwenden wollen, besitzen Sie nun eine alternative Möglichkeit. Sollten Sie jedoch annehmen, Ihre Routinen auf diese Weise beschleunigen zu können, muß ich Sie leider enttäuschen. Sie ersparen sich zwar mehrere Unterprogrammaufrufe, der Zeitgewinn ist jedoch im Vergleich zur äußerst aufwendigen OPEN-Routine vernachlässigbar gering.

Die Umgehung der »offiziellen« Routinen SETPAR/SETNAM hat zudem einen Nachteil, den Sie spätestens dann entdecken werden, wenn Sie in die Verlegenheit kommen sollten, Ihre Routinen auf einen anderen Commodore-Computer umschreiben zu müssen. Die Parameterübergabe an die Betriebssystemroutinen ist auf allen Commodore-Computern identisch. Dank der sogenannten »Kernel-Sprungtabelle« sind sogar die Einsprungadressen identisch.

Dieser Kompatibilitäts-Vorteil ist jedoch nur vorhanden, wenn Sie in Ihren Programmen die Betriebssystemroutinen wie vorgesehen benutzen und nicht wie gezeigt umgehen, indem Sie Werte direkt in die Zeropage schreiben. Die Zeropageadressen sind zum großen Teil von Rechner zu Rechner völlig unterschiedlich (Ausnahme: C 16, C 116 und Plus/4 mit identischer Zeropage-Belegung).

Sie kennen nun alle grundlegenden Betriebssystemroutinen zur Arbeit mit logischen Dateien. Im folgenden stelle ich Standardanwendungen wie die Abfrage des Fehlerkanals vor und gehe auf den Umgang mit Direktzugriffsdateien ein.

In den zugehörigen Programmbeispielen werden zwei Programmiertechniken verwendet, die Einsteiger in die Maschinenprogrammierung eventuell vor Probleme stellen und die ich daher kurz erläutere.

1. Sprung + RTS: Prinzipiell kann jede (!) Befehlsfolge JSR/RTS durch ein JMP ersetzt werden.

2. Unbedingter bedingter Sprung: Ebenfalls nicht jedem Einsteiger bekannt - jedoch eine »gängige« Methode - ist es, einen »JMP«-Befehl durch einen bedingten Sprung zu ersetzen, wenn der Zustand zumindest eines Flags mit Sicherheit bekannt ist. Zum Beispiel kann in der Befehlsfolge

```
LOOP1 ...
LOOP2 ...
      ...
      DEX
      BNE LOOP2
      JMP LOOP1
```

der Befehl »JMP LOOP1« durch den um ein Byte kürzeren Befehl »BEQ LOOP1« ersetzt werden, da das Zero-Flag an dieser Stelle des Programms immer (!) gesetzt ist.

Fehlerkanal abfragen

Bei der Arbeit mit einem Diskettenlaufwerk ist es unumgänglich, nach verschiedenen Aktionen den Fehlerkanal abzufragen. In Basic ist diese Abfrage ein Kinderspiel und das zugehörige Programm kennen wohl die meisten von Ihnen:

```
100 OPEN 1,8,15:REM FEHLERKANAL OEFFNEN
110 INPUT#1,A$,B$,C$,D$:REM MELDUNG EINLESEN
120 PRINT A$;B$;C$;D$:REM MELDUNG AUSGEBEN
130 CLOSE 1:REM FEHLERKANAL SCHLIESSEN
```

Mit den bisher behandelten Routinen müßte es möglich sein, diese Basic-Befehle durch ein Maschinenprogramm zu ersetzen. Zuerst wird eine logische Datei mit der Sekundäradresse 15 geöffnet und die Eingabe auf diese Datei umgeleitet. Anschließend kann die Fehlermeldung Zeichen für Zeichen mit BASIN eingelesen werden.

Das folgende kleine Programm liest den Fehlerkanal, wenn es mit SYS 49152 aufgerufen wird und gibt die jeweilige Fehlermeldung auf dem Bildschirm aus. Um das Programm zu verstehen, müssen Sie jedoch wissen, daß die Fehlermeldung durch ein <RETURN>, also den Code $0D, abgeschlossen wird.

```
. C000  LDA #$01     ;LOG.FILENUMMER
. C002  LDX #$08     ;GERAETEADRESSE
. C004  LDY #$0F     ;SEKUNDAERADRESSE 15
                      (=FEHLERKANAL)
. C006  JSR $FFBA    ;SETPAR AUFRUFEN
. C009  JSR $FFC0    ;OPEN AUFRUFEN
. C00C  LDX #$01     ;LOG.FILENUMMER
. C00E  JSR $FFC6    ;EINGABE AUF LOG.DATEI
```

```
. C011  JSR $FFCF        ;ZEICHEN LESEN (BASIN)
. C014  JSR $FFD2        ;ZEICHEN AUSGEBEN (BSOUT)
. C017  CMP #$0D         ;'RETURN'?
. C019  BNE $C011        ;NEIN => WEITERLESEN

. C01B  JSR $FFCC        ;STANDARDGERAETE SETZEN
. C01E  LDA #$01         ;LOG.FILENUMMER
. C020  JMP $FFC3        ;LOG.DATEI SCHLIESSEN U.
                          NACH BASIC
```

Diese Routine ist recht einfach. Vor dem Aufruf von OPEN werden wie üblich mit SETPAR die Dateiparameter gesetzt. Der Aufruf von SETNAM entfällt, da wir nicht auf eine physikalisch auf der Diskette vorhandene Datei zugreifen wollen (beachten Sie den Unterschied zwischen einer logischen Datei, die eher als »Kanal« bezeichnet werden kann und einer physikalisch vorhandenen Datei), ein Dateiname daher nicht benötigt wird.

Nach dem Öffnen der Datei und dem Umlenken der Eingabe mit CHKIN wird Zeichen für Zeichen der Fehlermeldung eingelesen und auf dem Bildschirm ausgegeben.

Wenn das Zeichen $0D (<RETURN>) gelesen wird, wurde die komplette Fehlermeldung eingelesen. Die Standardgeräte Bildschirm und Tastatur werden wieder gesetzt und die logische Datei geschlossen, bevor der Rücksprung nach Basic erfolgt.

Mit einem kleinen Basic-Programm läßt sich die Routine erproben:

```
100 OPEN 1,8,2,"GIBT ES NICHT,S,R"
110 CLOSE 1
120 SYS 49152:REM AUFRUF DER MASCHINENROUTINE
```

Wenn Sie dieses Programm (Sie finden es auch als MSE-Listing 2) starten und das File »GIBT ES NICHT« auch tatsächlich nicht auf der eingelegten Diskette vorhanden ist, erhalten Sie auf dem Bildschirm die Fehlermeldung »62, FILE NOT FOUND,00,00«.

```
Name : c64.fehlerk.ass   c000 c025
----------------------------------
c000 : a9 01 a2 08 a0 0f 20 ba  4c
c008 : ff 20 c0 ff a2 01 20 c6  88
c010 : ff 20 cf ff 20 d2 ff c9  3f
c018 : 0d d0 f6 20 cc ff a9 01  c4
c020 : 4c c3 ff 01 08 a9 ff 38  ac
```

Listing 2. Die Fehlerkanalabfrage in Maschinensprache

Beim praktischen Einsatz der Routine innerhalb eines größeren Maschinenprogramms sollten Sie die Fehlermeldung nicht direkt auf dem Bildschirm ausgeben, sondern in einen Pufferspeicher schreiben. Wenn die Meldung komplett eingelesen wurde, kann die Fehlernummer überprüft werden (die beiden ersten Bytes). Ist sie Null, trat kein Fehler auf, ansonsten kann die gesamte Meldung ausgegeben und – je nach Fehlernummer – entsprechend reagiert werden.

Prüfen, ob Gerät vorhanden (LISTEN und UNLISTEN)

Die Routine zur Abfrage des Fehlerkanals nützt leider nichts, wenn es darum geht, zu erkennen, ob ein bestimmtes Peripheriegerät überhaupt vorhanden ist (das heißt sowohl angeschlossen als auch eingeschaltet).

Das Problem kann gelöst werden, wenn wir uns etwas tiefer in die »Eingeweide« des Betriebssystems wagen, an die Routinen zur Behandlung des seriellen Busses, die von den »übergeordneten« Routinen zur Behandlung logischer Dateien natürlich ebenfalls verwendet werden.

Die Routine LISTEN ($FFB1) sendet an ein angegebenes Gerät einen sogenannten »LISTEN«-Befehl und teilt diesem damit mit, daß es sich zum Empfang von Daten bereithalten soll. Die Geräteadresse wird LISTEN im Akku übergeben.

Die UNLISTEN-Routine ($FFAE) bewirkt das Gegenteil. Sie beendet jede Datenübertragung und macht den seriellen Bus damit für folgende Datenübertragungen wieder frei.

Um abzufragen, ob ein bestimmtes Gerät vorhanden ist, müssen wir folgendermaßen vorgehen:

1. Statusvariable (in $90) löschen
1. Akku mit Geräteadresse laden
2. LISTEN aufrufen
3. UNLISTEN aufrufen
4. Status prüfen

Anstelle der Routine READST verwende ich zur Abwechslung einmal direkt die Statusvariable in $90. READST macht nichts anderes, als den Inhalt dieser Speicherstelle zu lesen und im Akku an das aufrufende Programm zu übergeben. Sie können daher den aktuellen Status auch jederzeit direkt durch Zugriff auf $90 überprüfen.

Teil 1: Prüfen, ob Gerät vorhanden ist

```
. C000  LDA #$00         ;STATUSVARIABLE $90
. C002  STA $90          ;MIT $00 INITIALISIEREN
. C004  LDA #$08         ;GERAETEADRESSE DER FLOPPY
. C006  JSR $FFB1        ;LISTEN SENDEN
. C009  JSR $FFAE        ;UNLISTEN SENDEN
. C00C  LDA $90          ;STATUSVARIABLE LESEN
. C00E  BEQ $C026        ;GERAET VORHANDEN? JA=>
```

Teil 2: Gerät ist nicht vorhanden: Fehlerbehandlung

```
. C010  LDX #$00         ;ZAEHLER INITIALISIEREN
. C012  LDA $A1D0,X      ;AUF 'DEVICE NOT PRESENT'
                          ZUGREIFEN
. C015  PHA              ;AKKU RETTEN
. C016  AND #$7F         ;BIT 7 AUSBLENDEN
. C018  JSR $FFD2        ;'DEVICE NOT PRESENT'
                          AUSGEBEN
. C01B  INX              ;ZAEHLER INKREMENTIEREN
. C01C  PLA              ;AKKU HOLEN
. C01D  BPL $C012        ;BIT 7 GESETZT? JA=>
. C01F  JSR $FFE4        ;GET AUFRUFEN (ANALOG
                          BASIC-GET)
. C022  BEQ $C01F        ;TASTE GEDRUECKT? NEIN=>
. C024  BNE $C000        ;IMMER SPRINGEN !
. C026  RTS              ;ZURUECK NACH BASIC
```

Nach dem Aufruf mit SYS 49152 prüft Teil 1 dieser Routine (siehe auch Listing 3) wie beschrieben, ob das jeweilige Gerät – in diesem Fall die Floppystation – vorhanden ist. Wenn ja, erfolgt der Rücksprung nach Basic.

```
Name : device-pres.ass   c000 c029
----------------------------------
c000 : a9 00 85 90 a9 08 20 b1  db
c008 : ff 20 ae ff a5 90 f0 16  92
c010 : a2 00 bd d0 a1 48 29 7f  3c
c018 : 20 d2 ff e8 68 10 f3 20  d5
c020 : e4 ff f0 fb d0 da 60 ff  25
c028 : ff f7 18 65 33 85 f8 90  3a
```

Listing 3. Prüft, ob die Floppystation eingeschaltet ist

Teil 2 der Routine ist ein etwas primitives Beispiel einer Fehlerbehandlungsroutine. Wenn die Floppystation nicht angeschlossen oder nicht eingeschaltet ist, wird die Meldung »DEVICE NOT PRESENT« auf dem Bildschirm ausgegeben. Diese Fehlermeldung befindet sich im ROM ab Adresse $A1D0. Das letzte Zeichen jeder ROM-Fehlermeldung wird durch ein gesetztes siebtes Bit gekennzeichnet, daher die merkwürdige »Pull-« und »Pusherei« in der Routine und das Ausblenden des siebten Bits vor der Zeichenausgabe mit BSOUT.

Wenn die Meldung ausgegeben wurde, wird die Routine GET ($FFE4) aufgerufen, die ein Zeichen von der Tastatur einliest und im Akku übergibt. Wenn keine Taste gedrückt wurde, übergibt GET eine Null und das Zero-Flag ist gesetzt. Bei gesetztem Zero-Flag wird wieder nach GET gesprungen, daher haben wir eine Eingabewarteschleife analog dem Basic-Befehl:

```
100 GET A$:IF A$="" THEN 100:REM AUF TASTE WARTEN
```

Wenn eine beliebige Taste gedrückt wurde, wird zum Programmanfang verzweigt und erneut geprüft, ob die Floppystation vorhanden ist. Sinn dieser Fehlerbehandlung ist es, nach Ausgabe der Meldung »DEVICE NOT PRESENT« dem Benutzer Gelegenheit zur Korrektur des Fehlers zu geben, bevor die Überprüfung nach Betätigung einer beliebigen Taste wiederholt wird.

Diese Fehlerbehandlung hat selbstverständlich nur Beispielcharakter und ist ein wenig unschön, da sich das Programm in einer Endlosschleife befindet, wenn es der Benutzer nicht schafft, den Fehler zu beheben.

Entscheidend ist der erste Programmteil, den Sie jederzeit in Ihren Programmen verwenden können. Wenn das Programm ein wenig flexibler gestaltet und der Akku vor dem Aufruf von LISTEN nicht immer mit dem Wert $08 geladen wird, kann das Vorhandensein jedes beliebigen Geräts mit dieser Routine überprüft werden.

Diese Routine eignet sich hervorragend zum Einbau in Basic-Programme mit professionellem Anspruch, die nicht »abstürzen« dürfen, wenn der Benutzer vergaß, Laufwerk oder Drucker einzuschalten.

Relative Dateien

Die Verwaltung relativer Dateien ist weit schwieriger als der Umgang mit sequentiellen Dateien und bereits in Basic nicht gerade einfach zu programmieren. Wenn Sie jedoch professionell mit Dateien arbeiten wollen, führt kein Weg an relativen Dateien vorbei, da diese Dateiart im Gegensatz zu den sequentiellen Dateien den unmittelbaren Zugriff auf bestimmte Datensätze gestattet.

Im folgenden gehe ich davon aus, daß Ihnen der Umgang mit relativen Dateien in Basic vertraut ist (wenn nicht: In diesem Sonderheft finden Sie einen Artikel, der ausführlich auf sequentielle und relative Dateien eingeht).

Um den Umgang mit relativen Dateien zu demonstrieren, werde ich ein kleines Basic-Programm verwenden, das eine relative Datei mit 50 Records à 50 Byte anlegt, Datensätze in diese Datei schreibt und wieder daraus liest. Anschließend werden die einzelnen Teile dieses Basic-Programms durch Maschinenroutinen ersetzt.

Basic-Demo: Relative Dateien

```
100 REM *** REL-DATEI ANLEGEN ***
110 OPEN 15,8,15:REM BEFEHLSKANAL OEFFNEN
120 R=50:GOSUB 330:GOSUB 410:PRINT#1,CHR$(255):
    GOSUB 370:R=1
130 :
140 REM *** DATENSAETZE SCHREIBEN ***
150 INPUT"DATEN (ENDE='E')";D$
160 IF D$="E" THEN 220
170 GOSUB 330:GOSUB 410:PRINT#1,D$:GOSUB 370
180 R=R+1
190 GOTO 150
200 :
210 REM *** DATENSAETZE LESEN ***
220 AD=R-1
230 GOSUB 330
240 :
250 FOR I=1 TO AD
260 : R=I:GOSUB 410:INPUT#1,D$:PRINT D$
270 NEXT
280 :
290 GOSUB 370:CLOSE 15
300 END
310 :
320 REM * REL-DATEI OEFFNEN *
330 OPEN 1,8,2,"REL-DATEI,L,"+CHR$(50)
340 RETURN
350 :
360 REM * REL-DATEI SCHLIESSEN *
370 CLOSE 1
380 RETURN
390 :
400 REM * POSITIONIEREN *
410 HB=INT(R/256):LB=R-HB*256
420 PRINT#15,"P"+CHR$(2)+CHR$(LB)+CHR$(HB)+CHR$(1)
430 RETURN
```

Deklarationsteil

Den ersten Teil des Source-Codes bildet die Deklaration der bereits bekannten Betriebssystemroutinen:

```
;***********************
;* REL-DATEI VERWALTEN *
;*     S.BALOUI/1986    *
;***********************
;
.BA $C000              ;STARTADRESSE
;
;* VERWENDETE BETRIEBSSYSTEMROUTINEN *
.EQ BSOUT   = $FFD2   ;ZEICHEN AUSGEBEN
.EQ BASIN   = $FFCF   ;ZEICHEN LESEN
.EQ OPEN    = $FFC0   ;LOG.DATEI OEFFNEN
.EQ CLOSE   = $FFC3   ;LOG.DATEI SCHLIESSEN
.EQ CKOUT   = $FFC9   ;AUSGABE AUF LOG.DATEI LEGEN
.EQ CHKIN   = $FFC6   ;EINGABE AUF LOG.DATEI LEGEN
.EQ CLRCH   = $FFCC   ;KANAELE 'SAEUBERN'
.EQ SETFLS  = $FFBA   ;FILEPARAMETER SETZEN
.EQ SETNAM  = $FFBD   ;FILENAME SETZEN
```

Unterprogramme

Das Programm selbst läßt sich am einfachsten verstehen, wenn Sie sich zuerst die verwendeten Unterprogramme anschauen:

```
;* REL-DATEI OEFFNEN *
RELAUF  LDA #$01       ;LOG.FILENUMMER
        LDX #$08       ;GERAETEADRESSE
        LDY #$02       ;SEKUNDAERADRESSE
        JSR SETPAR     ;FILEPAR.SETZEN

        LDA #$0D       ;FILENAME: 13 ZEICHEN
        LDX #<NAME     ;ADRESSE NAME LOW
        LDY #>NAME     ;ADRESSE NAME HIGH
        JSR SETNAM     ;FILENAME SETZEN

        JMP OPEN       ;LOG.DATEI OEFFNEN

;* REL-DATEI SCHLIESSEN *
RELZU   LDA #$01       ;LOG.FILENUMMER
        JMP CLOSE      ;LOG.DATEI SCHLIESSEN

;* POSITIONIEREN *
POSIT   LDX #$0F       ;LOG.FILENUMMER
        JSR CKOUT      ;AUSGABE AUF LOG.DATEI LEGEN
        LDX #$00       ;X-REG.INITIALISIEREN
POSIT1  LDA STRING,X   ;POSITIONIERBEFEHL
        JSR BSOUT      ;ZEICHENWEISE SENDEN
        INX
        CPX #$05       ;FERTIG?
        BNE POSIT1     ;NEIN =>
        INC STRING+2   ;=>WEIST AUF NAECHSTEN
                       RECORD
        JMP CLRCH      ;KANAELE 'SAEUBERN'+ RTS!

NAME    .TX "REL-DATEI,L,"
        .BY $32
STRING  .BY "P",$02,$32,$00,$01
DATEN   .EN            DATENPUFFER AM PROG.ENDE
```

```
Name : c64.rel-datei.as  c000 c0c2
-----------------------------------
c000 : a9 0f a8 a2 08 20 ba ff  1c
c008 : 20 c0 ff 20 7c c0 20 96  08
c010 : c0 a2 01 20 c9 ff a9 ff  a9
c018 : 20 d2 ff 20 91 c0 a9 01  6d
c020 : 8d bd c0 a2 00 20 cf ff  50
c028 : 9d c0 c0 e8 c9 0d d0 f5  a7
c030 : e0 02 f0 1e 20 7c c0 20  3a
c038 : 96 c0 a2 01 20 c9 ff a2  8d
c040 : 00 bd c0 c0 20 d2 ff e8  d1
c048 : c9 0d d0 f5 20 91 c0 4c  b5
c050 : 23 c0 ca 8e bd c0 20 7c  b3
c058 : c0 20 96 c0 a2 01 20 c6  26
c060 : ff 20 cf ff c9 ff f0 09  d5
c068 : 20 d2 ff c9 0d d0 f2 f0  2f
c070 : e8 20 cc ff 20 91 c0 a9  80
c078 : 0f 4c c3 ff a9 01 a2 08  db
c080 : a0 02 20 ba ff a9 0d a2  47
c088 : ae a0 c0 20 bd ff 4c c0  49
c090 : ff a9 01 4c c3 ff a2 0f  13
c098 : 20 c9 ff a2 00 bd bb c0  4f
c0a0 : 20 d2 ff e8 e0 05 d0 f5  ac
c0a8 : ee bd c0 4c cc ff 52 45  cf
c0b0 : 4c 2d 44 41 54 45 49 2c  b9
c0b8 : 4c 2c 32 50 02 32 00 01  65
c0c0 : e1 00 00 00 00 00 00 00  a2
```

Listing 4. Ein Beispiel für die Arbeit mit relativen Dateien in Maschinensprache

Dieses Programm finden Sie auch in Listing 4. Das Öffnen der relativen Datei unterscheidet sich nicht im geringsten vom Öffnen einer sequentiellen Datei. Zuerst werden SETPAR die Fileparameter übergeben, danach wird SETNAM die Länge des Filenamens und ein Zeiger auf die Adresse des Namens übergeben, anschließend wird die OPEN-Routine aufgerufen.

Das Schließen der Datei wird ebenfalls wie gewohnt vorgenommen, indem der Routine CLOSE die logische Filenummer im Akku übergeben wird.

Interessanter ist die Positionierroutine. Zu Beginn des Hauptprogramms wird der Befehlskanal unter Angabe der logischen Filenummer geöffnet. Vor dem Positionieren auf einen bestimmten Record muß die Ausgabe auf diese logische Datei umgelenkt werden. Anschließend wird der Befehlsstring an die Floppystation gesendet und mit CLRCH die Standardgeräte gesetzt.

Anlegen der relativen Datei

Das eigentliche Hauptprogramm muß zuerst die relative Datei anlegen:

```
;* BEFEHLSKANAL OEFFNEN *
        LDA #$0F      ;LOG.FILENUMMER
        TAY           ;SEKUNDAERADRESSE
        LDX #$08      ;GERAETEADRESSE
        JSR SETPAR    ;FILEPAR.SETZEN
        JSR OPEN      ;BEFEHLSKANAL OEFFNEN
;
;* RECORD NR.50 FREIGEBEN *
        JSR RELAUF    ;REL-DATEI OEFFNEN
        JSR POSIT     ;AUF RECORD 'STRING+2'
                       POSITIONIEREN
        LDX #$01      ;AUSGABE AUF
        JSR CKOUT     ;REL-DATEI LEGEN
        LDA #$FF      ;FREIGABEKENNZEICHEN $FF
        JSR BSOUT     ;IN RECORD SCHREIBEN
        JSR RELZU     ;REL-DATEI SCHLIESSEN=
                        >ANLEGEN
```

Der Befehlskanal wird unter Angabe der logischen Filenummer 15 geöffnet, das Unterprogramm RELAUF zum Öffnen der relativen Datei aufgerufen und anschließend das Unterprogramm POSIT zum Positionieren auf Record Nummer 50.

Beachten Sie bitte, daß das Low-Byte der Recordnummer im Befehlsstring den Wert $32 enthält, exakt die gewünschte Recordnummer, in die nun das Freigabekennzeichen $FF geschrieben werden soll. Zuvor wird die Ausgabe auf die logische Datei mit der Filenummer $01 gelegt (die Datei »REL-DATEI«). Wenn der Wert $FF ausgegeben wurde, wird die relative Datei mit der Routine RELZU wieder geschlossen. Die Datei wird nun von der Floppystation selbständig angelegt werden.

Datensätze in eine relative Datei schreiben

Bevor der erste Datensatz in die Datei geschrieben werden kann, muß dafür gesorgt werden, daß POSIT auf den gewünschten Record 1 positioniert. Die Recordnummer ist das dritte Byte des Befehlsstrings (STRING+2). Dieses Byte erhält den Wert Eins.

```
;* DATENSAETZE SCHREIBEN *
        LDA #$01      ;RECORDNUMMER MIT 1
        STA STRING+2  ;INITIALISIEREN
```

Der erste Aufruf von POSIT führt daher zum Positionieren auf den ersten Record der relativen Datei. Da die Routine POSIT das Low-Byte der Recordnummer nach erfolgter Positionierung inkrementiert, führt der zweite Aufruf von POSIT dazu, daß auf Record 2 positioniert wird, der dritte Datensatz wird demnach in Record 3 geschrieben werden und so weiter.

Der erste Datensatz kann nun vom Benutzer eingegeben werden. Zum Einlesen verwendet das Programm die Betriebssystemroutine BASIN. Ebenso wie beim Basic-Befehl INPUT blinkt der Cursor und der Benutzer kann seine Eingabe beliebig editieren, bevor er sie mit <RETURN> abschließt.

```
WRITE1  LDX #$00      ;BENUTZEREINGABEN
WRITE2  JSR BASIN     ;HOLEN UND
        STA DATEN,X   ;IM PUFFER 'DATEN,X'
        INX           ;SPEICHERN
        CMP #$0D      ;EINGABEENDE?
        BNE WRITE2    ;NEIN =>
        CPX #$02      ;EINGABELAENGE 1 (+RETURN)?
        BEQ READ1     ;JA (=ENDE EINGABE) =>
```

Die eingelesenen Zeichen werden von der Routine in dem Puffer DATEN,X gespeichert, der sich am Programmende befindet. Die Routine liest Zeichen für Zeichen mit BASIN ein und speichert die Zeichen in diesem Puffer, bis der Code $0D (Carriage Return) gelesen wird.

Der nun folgende Programmteil wird übersprungen, wenn das X-Register nach dem Einlesen den Wert Zwei besitzt, das heißt, wenn außer Carriage Return nur ein weiteres Zeichen eingelesen wurde. Der Sinn dieser Überprüfung:

Nachdem ein vom Benutzer eingegebener String in der relativen Datei gespeichert wurde, soll der Benutzer ebenso wie in dem Basic-Demoprogramm beliebige weitere Datensätze eingeben können. Das Basic-Programm erkannte an der Eingabe »E«, daß die Eingabe beendet und die Sätze wieder aus der Datei eingelesen werden sollten. Diese »Abbruchbedingung« wurde von mir im Maschinenprogramm leicht abgewandelt. Das Eingabeende wird an der Stringlänge eins erkannt. Wenn Sie keine weiteren Datensätze mehr eingeben wollen, können Sie daher ein beliebiges Zeichen – unter anderem auch »E« (wie im Basic-Programm) eingeben, um die Datensatzeingabe zu beenden.

```
        JSR RELAUF    ;REL-DATEI OEFFNEN
        JSR POSIT     ;AUF 'STRING+2'POSITIONIEREN
        LDX #$01      ;AUSGABE AUF
        JSR CKOUT     ;REL-DATEI LEGEN

        LDX #$00      ;X INITIALISIEREN
WRITE3  LDA DATEN,X   ;DATEN ZEICHENWEISE AUS
        JSR BSOUT     ;PUFFER HOLEN UND AUF
        INX           ;REL-DATEI AUSGEBEN
```

```
        CMP #$0D       ;DATENSATZENDE?
        BNE WRITE3     ;NEIN =>
        JSR RELZU      ;REL-DATEI SCHLIESSEN
        JMP WRITE1     ;=> ANFANG EINGABE/
                        SCHREIBSCHLEIFE
```

Die relative Datei wird nun geöffnet, mit POSIT auf Record 1 positioniert (erinnern Sie sich: Vor der Eingabe des ersten Satzes wird STRING+2, das Low-Byte der Recordnummer, mit Eins initialisiert), und die Ausgabe wird auf die relative Datei gelegt.

Die Daten werden aus dem Puffer geholt und mit BASIN auf die Datei ausgegeben, wobei das Stringende an dem Code $0D (Carriage Return) erkannt wird. Die Datei wird geschlossen und der Beginn der »Schreibroutine« angesprungen, die Datensatzeingabe des Benutzers. Wie bereits erwähnt, wird diese Schleife wiederholt, bis ein String in der Länge Eins (plus Carriage Return) eingegeben wurde.

Wenn Sie sich wundern sollten, daß die Datei wegen jedem einzelnen Datensatz geöffnet und geschlossen wird: Es ist möglich, die relative Datei einmalig vor Beginn der Routine zum Schreiben von Sätzen zu öffnen und erst dann zu schließen, wenn der letzte Datensatz eingetragen wurde. Diese Vorgehensweise beinhaltet jedoch ein Sicherheitsrisiko. Das Betriebssystem sammelt Daten, die in eine logische Datei geschrieben werden, in einem Puffer. Der Inhalt dieses Puffers wird erst dann tatsächlich in den jeweiligen Record übertragen, wenn der Puffer entweder voll ist oder aber wenn die Datei – wie im Beispiel nach jedem Satz – geschlossen wird. Ich empfehle Ihnen, bei Schreibzugriffen auf eine relative Datei diese nach jedem Satz zu schließen. Dadurch wird gewährleistet, daß zum Beispiel bei einem Stromausfall niemals mehrere im Puffer enthaltene Datensätze verlorengehen können, sondern nur der gerade bearbeitete Satz.

Datensätze aus einer relativen Datei lesen

Bevor die Daten nun wieder eingelesen werden können, muß zuerst auf den ersten Record der Datei positioniert werden (Das X-Register enthält nach Verlassen der Leseroutine den Wert Zwei):

```
;* DATENSAETZE LESEN *
READ1   DEX            ;X WAR 2, NUN 1 !
        STX STRING+2   ;RECORDNUMMER MIT 1
                        INITIALISIEREN
```

Die relative Datei wird geöffnet, auf Record 1 positioniert und die Eingabe auf die Datei gelegt:

```
        JSR RELAUF     ;REL-DATEI OEFFNEN
READ2   JSR POSIT      ;AUF 'STRING+2'POSITIONIEREN
        LDX #$01       ;EINGABE AUF
        JSR CHKIN      ;REL-DATEI LEGEN
```

Die Datensätze können nun mit BASIN zeichenweise eingelesen und mit BSOUT auf dem Bildschirm ausgegeben werden, wobei der Code $0D das Ende eines Datensatzes markiert. Wurde ein Datensatz komplett eingelesen, erfolgt ein Sprung nach READ2. Anschließend wird auf den nächsten Record positioniert und dieser eingelesen:

```
READ3   JSR BASIN      ;EINGABE VON REL-DATEI
        CMP #$FF       ;$FF(=ERSTER UNBELEGTER
                        RECORD)?
        BEQ READ4      ;JA (=DATEIENDE) =>
        JSR BSOUT      ;ZEICHEN AUF SCREEN AUSGEBEN
        CMP #$0D       ;CARR.RETURN (=SATZENDE)?
        BNE READ3      ;NEIN =>
        BEQ READ2      ;IMMER SPRINGEN !
```

$FF wird eingelesen, wenn auf den ersten nicht belegten Record positioniert wurde, das heißt, wenn alle belegten Records bereits gelesen und ausgegeben wurden (Erinnern Sie sich: Mit $FF werden alle Records beim Anlegen der Datei automatisch von der Floppystation als nicht belegt gekennzeichnet). In diesem Fall werden mit CLRCH die Standardgeräte gesetzt und die geöffneten Dateien mit CLOSE geschlossen, bevor die Rückkehr nach Basic erfolgt:

```
READ4   JSR CLRCH      ;KANAELE 'SAEUBERN',
                        STANDARDGERAETE
        JSR RELZU      ;REL-DATEI SCHLIESSEN
        LDA #$0F       ;BEFEHLSKANAL SCHLIESSEN
        JMP CLOSE      ;UND NACH BASIC (RTS !)
```

Dieses Demoprogramm ist in mehrfacher Hinsicht sehr einfach gestaltet, da sowohl beim Schreiben als auch beim Lesen der Reihe nach auf unmittelbar folgende Records positioniert wird (Record 1, Record 2, Record 3 etc.). Die Positionierung wird sehr einfach, denn bei jedem Aufruf von POSIT kann der Zähler für die Recordnummer einfach inkrementiert werden. In der Praxis kommt dieser Idealfall leider nur selten vor.

Zum zweiten wird im Beispiel nur das Low-Byte der Recordnummer verwendet (STRING+2), während das High-Byte (STRING+3) immer den Wert Null enthält. Bei Dateien mit mehr als 255 Records müssen Sie selbstverständlich auch das High-Byte entsprechend setzen.

In der Praxis werden Sie auf eine relative Datei meist über eine im Speicher gehaltene Indexdatei zugreifen, die die Recordnummer und beispielsweise den Namen des zugehörigen Datensatzes enthält. Der Benutzer sucht Herrn »Müller«, Ihr Programm durchsucht die Indexdatei nach »Müller« und findet zum Beispiel die zugehörige Recordnummer $20/$03 (Low-Byte/High-Byte). Auf diesen Record könnten Sie folgendermaßen positionieren:

```
...                 ;DURCHSUCHEN DER INDEXDATEI
...                 ;NACH AUFFINDEN VON 'MUELLER'
...                 ;WEIST DER ZEIGER POINTR(+1) AUF
LDA (POINTR),Y      ;DIE ZUGEHOERIGE RECORDNUMMER
STA STRING+2        ;LOW-UND HIGH-BYTE DER
INY                 ;RECORDNUMMER WERDEN NACH
LDA (POINTR),Y      ;STRING+2 UND STRING+2 UEBERTRAGEN
STA STRING+3        ;ANSCHLIESSEND WIRD DIE
JSR POSIT           ;POSITIONIERROUTINE AUFGERUFEN
...
...
```

Auf diese Weise dürfte die Positionierung in einer vollständigen Dateiverwaltung ablaufen. Zweck des Demoprogramms ist nicht die Praxisnähe, sondern allein die Darstellung der relevanten Prinzipien.

Verwenden Sie zur Eingabe des Programms das MSE-Listing 4. Zum Testen rufen Sie das Programm mit SYS 49152 auf. Geben Sie mehrere Datensätze ein. Beenden Sie die Eingabe durch die Eingabe eines Satzes, der aus einem einzigen Zeichen besteht (»E« oder »X« etc.). Die Sätze werden nun der Reihe nach aus der relativen Datei eingelesen.

Achten Sie bei der Behandlung relativer Dateien generell auf folgende Punkte:

1. Beim Schreiben in eine relative Datei sind sowohl die Datei selbst als auch der Befehlskanal geöffnet, und Daten müssen abwechselnd über den Datei- und den Befehlskanal gesendet werden. Die Ausgabe muß daher ständig mit CKOUT umgelenkt werden. Vergessen Sie in solchen Fällen bitte keinesfalls, vor dem Aufruf von CKOUT die Übertragungskanäle durch den Aufruf von CLRCH zu »säubern«.

2. Wenn Sie an einer Routine arbeiten und allmählich verzweifeln, weil die Aus- beziehungsweise Eingaben sich prinzipiell auf die falsche Datei beziehen, liegt die Ursache – zumindest bei meinen eigenen Programmen – meist in einem noch gültigen, vom Programmierer jedoch inzwischen vergessenen früheren Aufruf von CKOUT oder CHKIN.

3. Wenn Datensätze wie im Demoprogramm in unmittelbar aufeinanderfolgende Records geschrieben werden, müssen

Sie nur beim ersten Datensatz auf den Anfangsrecord positionieren. Die Floppystation positioniert beim Schreiben von Daten in Record n selbständig auf den jeweils nächsten Record +1. Im Demoprogramm hätte daher beim Schreiben der Datensätze auf das Inkrementieren der Recordnummer verzichtet werden können.

Beachten Sie bitte, daß diese »automatische Inkrementierung« nur beim Schreiben, nicht jedoch beim Lesen aus einer relativen Datei vorgenommen wird!

Die Direktzugriffsbefehle B-P, U1 und U2

Relative Dateien sind zwar recht nützlich, jedoch nicht unbedingt die schnellste Art und Weise zur Verwaltung einer Direktzugriffsdatei. Die Ursache liegt in der nötigen zweimaligen – und sehr gemächlichen – Kopfpositionierung der Floppystation. Zuerst wird auf die sogenannten »Side-Sektoren« positioniert, die eine Tabelle aller Records mit ihrer jeweiligen Position (Spur und Sektor) enthalten. Erst anschließend wird auf den gewünschten Record zugegriffen.

Erheblich schneller ist die Verwaltung von Direktzugriffsdateien mit den »Direktzugriffsbefehlen«, da hier durch Angabe von Spur und Sektor sofort auf einen Datensatz zugegriffen wird.

Nur mit diesen Befehlen ist es möglich, auf die Directory zuzugreifen, um den Diskettennamen oder die ID zu verändern, gelöschte Files wiederherzustellen und andere Manipulationen vorzunehmen.

Aus Platzgründen muß ich im folgenden wieder davon ausgehen, daß Ihnen diese Befehle bereits in Basic vertraut sind und werde mich auf knappe Erläuterungen beschränken. Die Verbindung zwischen Basic und Maschinensprache stelle ich anhand eines kleinen Demoprogramms her, das die Änderung der Disketten-ID erlaubt.

ID ändern (Basic)

```
100 OPEN 15,8,15:REM BEFEHLSKANAL OEFFNEN
110 OPEN 2,8,2,"#":REM DATENKANAL OEFFNEN
120 PRINT#15,"U1:"2;0;18;0:REM SPUR 18, SEKTOR 0
    IN PUFFER
130 PRINT#15,"B-P:"2;162:REM PUFFER-POINTER AUF
    BYTE NR.162
140 PRINT "ALTE ID: ";
150 GET#2,A$:PRINT A$:GET#2,A$:PRINT A$:REM
    ID LESEN/AUSGEBEN
160 :
170 INPUT"NEUE ID";ID$:REM BENUTZEREINGABE DER
    NEUEN ID
180 PRINT#15,"B-P:"2;162:REM PUFFER-POINTER WIEDER
    AUF BYTE 162
190 PRINT#2,LEFT$(ID$,2);:REM ID-BYTES
    UEBERSCHREIBEN
200 PRINT#15,"U2:"2;0;18;0:REM PUFFERINHALT
    ZURUECKSCHREIBEN
210 PRINT#15,"I":REM FLOPPY INITIALISIEREN
220 CLOSE 2:CLOSE 15:REM KANAELE SCHLIESSEN
```

Beim Öffnen des Datenkanals wird als Dateiname »#« angegeben. Der Grund: Durch dieses Zeichen reserviert die Floppystation intern einen freien Pufferspeicher für die Datenübertragung. Sie könnte durch »#0«, »#1«, »#2« oder »#3« auch gezwungen werden, einen bestimmten Puffer zu reservieren, was jedoch nicht empfehlenswert ist.

Die ID befindet sich in Sektor 0 von Spur 18. Dieser Block wird mit »U1« von der Diskette gelesen und befindet sich zunächst im reservierten Puffer der Floppystation. Der Pufferpointer wird mit »B-P« auf das erste Byte der ID gesetzt (Byte 162), das erste und zweite Byte der ID werden mit GET# aus dem Puffer gelesen und auf dem Bildschirm ausgegeben.

Der Benutzer kann nun eine neue ID eingeben, die dem String ID$ zugewiesen wird. Der Pufferpointer wird wieder auf das erste Byte der ID gesetzt (beim Einlesen aus dem Puffer wurde er verändert). Die beiden ersten Zeichen der neu eingegebenen ID werden über den Datenkanal gesendet und die alte ID damit überschrieben. Die Änderung wird erst wirksam, wenn der Inhalt des Puffers mit »U2« auf die Diskette zurückgeschrieben wird.

Im Floppy-RAM befindet sich die unveränderte Directory. Wenn Sie sich die Directory anschauen, sehen Sie daher unverändert die alte ID. Zum Abschluß wird deshalb der Initialisierungsbefehl »I« an die Floppystation gesendet, der das Neueinlesen der Directory von der eingelegten Diskette bewirkt. Erst jetzt »weiß« die Floppystation die geänderte ID.

Übrigens: Dieses Programm verwendet für die ID nur zwei Zeichen, ebenso wie es im Floppy-Handbuch beschrieben ist. Tatsächlich kann die ID bis zu fünf Zeichen lang sein (probieren Sie's aus und ändern Sie das Basic-Programm in Zeile 190 entsprechend).

ID ändern (Maschinensprache)

In Maschinensprache ist diese Aktion leider ungleich aufwendiger und verlangt nach einem recht langen Programm (zumindest bei mir, vielleicht können Sie es besser). Da ein Disassemblerlisting dieses Programms zu unübersichtlich wäre, wird die Darstellung als Assembler-Sourcecode verwendet. Listing 5 zeigt den fertigen Objectcode.

```
Name : c64.id-change      c000 c0fd
-----------------------------------
c000 : 20 e7 ff a9 0f a8 a2 08   1a
c008 : 20 ba ff 20 c0 ff a9 02   40
c010 : a8 a2 08 20 ba ff a9 01   64
c018 : a2 a4 a0 c0 20 bd ff 20   7d
c020 : c0 ff a9 c9 a0 c0 20 86   21
c028 : c0 a9 af a0 c0 20 97 c0   aa
c030 : a9 a5 a0 c0 20 97 c0 a2   f3
c038 : 02 20 c6 ff 20 cf ff 20   bd
c040 : d2 ff 20 cf ff 20 d2 ff   60
c048 : a9 0d 20 d2 ff a9 a5 a0   ff
c050 : c0 20 97 c0 a9 d3 a0 c0   5b
c058 : 20 86 c0 a2 02 20 c9 ff   88
c060 : 20 cf ff 20 d2 ff 20 cf   b9
c068 : ff c9 0d d0 02 a9 a0 20   d9
c070 : d2 ff a9 bd a0 c0 20 97   24
c078 : c0 a2 0f 20 c9 ff a9 49   27
c080 : 20 d2 ff 4c e7 ff 85 f0   09
c088 : 84 f1 a0 00 b1 f0 f0 06   9f
c090 : 20 d2 ff c8 d0 f6 60 48   09
c098 : a2 0f 20 c9 ff 68 20 86   d4
c0a0 : c0 4c cc ff 23 42 2d 50   53
c0a8 : 3a 32 2c 31 36 32 00 55   cc
c0b0 : 31 3a 32 2c 30 2c 31 38   aa
c0b8 : 2c 30 00 00 ff 55 32 3a   e4
c0c0 : 32 2c 30 2c 31 38 2c 30   80
c0c8 : 00 41 4c 54 45 20 49 44   09
c0d0 : 20 3a 00 4e 45 55 45 20   2c
c0d8 : 49 44 20 3f 00 20 30 30   56
c0e0 : 33 30 30 33 33 33 30 30   8c
c0e8 : 00 30 30 32 32 30 30 30   19
c0f0 : 30 32 00 30 30 32 30 30   f5
c0f8 : 33 30 30 00 00 00 00 00   50
```

Listing 5. Ändern der Disketten-ID

```
;*******************
;* DISK-ID AENDERN *
;*  S.BALOUI/1986  *
;*******************
;
.BA $C000            ;PROGRAMMSTART
;
;* BETRIEBSSYSTEMROUTINEN/ZEROPAGEADRESSEN *
.EQPOINTR    = $F0   ;POINTER F.INDIR.INDIZ.
                      ADRESSIERG.
```

```
.EQ SETPAR   = $FFBA ;FILEPAR.SETZEN
.EQ SETNAM   = $FFBD ;FILENAME SETZEN
.EQ OPEN     = $FFC0 ;LOG.DATEI OEFFNEN
.EQ CLALL    = $FFE7 ;SCHLIESST ALLE (!) DATEIEN
.EQ CLRCH    = $FFCC ;KANAELE 'SAUEBERN'/
                      STANDARDDEVICES
.EQ CHKIN    = $FFC6 ;EINGABE AUF LOG.DATEI LEGEN
.EQ CKOUT    = $FFC9 ;AUSGABE AUF LOG.DATEI LEGEN
.EQ BSOUT    = $FFD2 ;ZEICHEN AUSGEBEN
.EQ BASIN    = $FFCF ;ZEICHEN EINLESEN
;
    JSR CLALL        ;OFFENE DATEIEN SCHLIESSEN
```

Die verwendeten Betriebssystemroutinen dürften Ihnen nun bekannt sein. Neu hinzu kommt in diesem Deklarationsteil die Routine CLALL, die ebenso wie CLRCH die Kanäle bereinigt und alle (!) offenen Dateien schließt. CLALL sollte zur Vermeidung eines »FILE OPEN ERROR« am Beginn jedes Programms aufgerufen werden.

```
;* BEFEHLS- UND DATENKANAL OEFFNEN *
        LDA #$0F        ;BEFEHLSKANAL OEFFNEN
        TAY             ;(LF=15, SA=15, GA=8)
        LDX #$08        ;KEIN NAME ERFORDERLICH
        JSR SETPAR
        JSR OPEN
;
        LDA #$02        ;DATENKANAL OEFFNEN
        TAY             ;(LF=2, SA=2, GA=8)
        LDX #$08        ;
        JSR SETPAR      ;ALS DATEINAME WIRD
        LDA #$01        ;'#'VERWENDET (AM
        LDX #<NAME      ;PROGRAMMENDE ABGELEGT)
        LDY #>NAME
        JSR SETNAM
        JSR OPEN
```

Das Öffnen von Befehls- und Datenkanal verläuft äquivalent zur Behandlung relativer Dateien. Der Filename wurde ebenso wie verschiedene andere Strings am Programmende abgelegt.

```
;* AKTUELLE ID LESEN UND AUSGEBEN *
    LDA #<ALT     ;DEM UNTERPROG. 'STROUT' WIRD EIN
    LDY #> ALT    ;POINTER AUF 'ID ALT:' UEBERGEBEN
    JSR STROUT    ;AUSGABE VON 'ID ALT:'AUF SCREEN
    LDA #<LIES    ;POINTER AUF DEN BLOCK-READ-
                   STRING
    LDY #>LIES    ;("U1:2,0,18,0") UND STRING-
                   AUSGABE
    JSR BEFEHL    ;AUF BEFEHLSKANAL
    LDA #<BP      ;POINTER AUF PUFFER-POINTER-
                   BEFEHL
    LDY #>BP      ;("B-P:2,162") UND AUSGABE AUF
    JSR BEFEHL    ;BEFEHLSKANAL
```

Am Programmende befinden sich die Unterprogramme STROUT und BEFEHL. STROUT gibt einen String aus, auf den in Akku und Y-Register ein Pointer übergeben wird. Vor dem Einlesen der ID wird mit STROUT der String »ID ALT :« auf dem Bildschirm ausgegeben.

Zum Einlesen der ID muß zuerst der »U1«-Befehl und anschließend der »B-P«-Befehl über den Befehlskanal gesendet werden. Das Unterprogramm BEFEHL leitet die Ausgabe auf die logische Datei $0F, den Befehlskanal, um, gibt den String aus, auf den mit Akku und Y-Register ein Pointer übergeben wurde, und ruft zuletzt CLRCH auf, um die Kanäle zu »bereinigen« und die Standardein-/ausgabegeräte zu setzen.

```
        LDX #$02        ;EINGABE AUF
        JSR CHKIN       ;DATENKANAL LEGEN
        JSR BASIN       ;DIE BEIDEN ZEICHEN DER
        JSR BSOUT       ;ID AUS DEM PUFFER
        JSR BASIN       ;EINLESEN UND AUF
        JSR BSOUT       ;DEM BILDSCHIRM AUSGEBEN
        LDA #$0D        ;ZUM ABSCHLUSS ZEILEN-
                         VORSCHUB
        JSR BSOUT       ;AUF SCREEN AUSGEBEN
```

Nachdem die Eingabe auf den Datenkanal umgeleitet wurde, wird die ID eingelesen und auf dem Bildschirm ausgegeben. Daß zum Abschluß ein Zeilenvorschub auf dem Bildschirm ausgegeben wird, hat nur den Zweck, die folgende Ausgabe von »NEUE ID ?« in der nächsten Bildschirmzeile beginnen zu lassen.

```
;* NEUE ID EINLESEN UND DIRECTORY AENDERN *
        LDA #<BP        ;"B-P"-BEFEHLSSTRING
        LDY #>BP        ;AUF BEFEHLSKANAL
        JSR BEFEHL      ;AUSGEBEN
        LDA #<NEU       ;STRING 'NEUE ID ?'
        LDY #>NEU       ;AUF SCREEN AUSGEBEN
        JSR STROUT
```

Bevor auf die Eingabe der neuen ID durch den Benutzer gewartet wird, wird der Pufferpointer wieder auf das Byte 162 gesetzt, indem der entsprechende Befehlsstring ausgegeben wird. Anschließend wird die Aufforderung »NEUE ID ?« auf dem Bildschirm ausgegeben.

```
        LDX #$02        ;AUSGABE AUF
        JSR CKOUT       ;DATENKANAL SETZEN
        JSR BASIN       ;2 ZEICHEN VON DER
        JSR BSOUT       ;TASTATUR EINLESEN
        JSR BASIN       ;UND AUF BEFEHLSKANAL
        CMP #$0D        ;AUSGEBEN
        BNE IDWRIT1     ;ALS 2.ZEICHEN 160
        LDA #160        ;(INVERSES SPACE) SENDEN,
IDWRIT1 JSR BSOUT       ;WENN NUR 1 ZEICHEN EINGEG.
```

Die Ausgabe wird nun auf den Datenkanal umgeleitet und zwei Zeichen – die neue ID –, die von der Tastatur eingelesen werden, auf den Datenkanal ausgegeben. Wenn der Benutzer nur ein Zeichen als ID eingab und danach die Eingabe mit <RETURN> beendete, liest BASIN als zweites Zeichen den Code $0D (=Carriage Return).

In diesem Fall wird ein inverses Leerzeichen (Code 160) als zweites Zeichen der ID verwendet und auf den Datenkanal ausgegeben. Warum ausgerechnet ein inverses Space benutzt wird, erkennen Sie sofort, wenn Sie sich ein beliebiges Directory anschauen: Die »Kopfleiste« des Directory ist mit inversen Spaces aufgefüllt.

```
        LDA #<WRITE     ;'U2'-BEFEHLSSTRING UEBER
        LDY #>WRITE     ;BEFEHLSKANAL AUSGEBEN UND
        JSR BEFEHL      ;DAMIT PUFFERINHALT ZURUECK-
        LDX #$0F        ;SCHREIBEN
        JSR CKOUT       ;INITIALISIERUNGSBEFEHL
        LDA #"I"        ;AN FLOPPY SENDEN, UM
                         DIRECTORY
        JSR BSOUT       ;ZU AKTUALISIEREN
;
        JMP CLALL       ;DATEIEN SCHLIESSEN + BASIC!
```

Die geänderte ID befindet sich zwar nun im Puffer, in dem Sektor 0 von Spur 18 eingelesen wurde. Der neue Pufferinhalt muß jedoch mit dem »U2«-Befehl auf die Diskette zurückgeschrieben werden, um die Änderung wirksam werden zu lassen.

Da sich im Floppy-RAM jedoch immer noch das alte Directory befindet, wird der Initialisierungsbefehl »I« gesendet. Bei der Initialisierung liest die Floppystation das (geänderte!) Directory ein. Wenn Sie nun das Directory laden, wird die geänderte ID ausgegeben.

Den Abschluß der Routine bildet das Bereinigen der Kanäle und Schließen aller offenen Dateien durch die Betriebssystemroutine CLALL.

Zum Verständnis dieses Programms fehlt noch die Darstellung der verwendeten Unterprogramme und Strings:

```
;* VERWENDETE STRINGS *
NAME     .BY "#"              ;SCHEINARGUMENT
BP       .TX "B-P:2,162"      ;PUFFER-POINTER SETZEN
         .BY $00
LIES     .TX "U1:2,0,18,0"    ;BLOCK LESEN
         .BY $00
WRITE    .TX "U2:2,0,18,0"    ;BLOCK SCHREIBEN
         .BY $00
ALT      .TX "ALTE ID :"
         .BY $00
NEU      .TX "NEUE ID ?"
         .BY $00
;
.EN                           ;PROGRAMMENDE
```

Alle mit STROUT beziehungsweise BEFEHL auszugebenden Strings schließen mit dem Byte $00 als »Endemarke« ab. Die in den Befehlen »B-P«, »U1« und »U2« anzugebenden Parameter müssen (!) bei Verwendung eines Assemblers als String abgelegt (bei Hypra-Ass mit dem Befehl ».TX«) und mit Kommata voneinander getrennt werden.

```
;* STRINGAUSGABE *
STROUT  STA POINTR        ;IN AKKU UND Y UEBERGEBEN
        STY POINTR+1      ;ZEIGER NACH POINTR(+1)
        LDY #$00          ;Y INITIALISIEREN
STR1    LDA (POINTR),Y    ;STRINGZEICHEN HOLEN
        BEQ STR2          ;ENDEMARKE $00 ERREICHT?
                           JA=>
        JSR BSOUT         ;SONST AUSGEBEN
        INY               ;ZEIGER AUF NEAECHSTES
                           ZEICHEN
        BNE STR1          ;IMMER SPRUNG !
STR2    RTS               ;ENDE DER STRINGAUSGABE
;
;* BEFEHLSAUSGABE *
BEFEHL  PHA               ;HIGH-BYTE DES STRINGZEIGERS
                           RETTEN
        LDX #$0F          ;AUSGABE AUF BEFEHLSKANAL
                           LEGEN
        JSR CKOUT         ;('CKOUT' AENDERT
                           AKKUINHALT!)
        PLA               ;HIGH-BYTE DES STRINGZEIGERS
                           HOLEN
        JSR STROUT        ;STRINGAUSGABE AUFRUFEN
        JMP CLRCH         ;STANDARDDEVICES + RTS !
```

Die Routine STROUT, die einen beliebigen String auf dem aktuellen Ausgabegerät ausgibt, sollte Ihnen keine Verständnisschwierigkeiten bereiten. Um die Routine BEFEHL zu verstehen, müssen Sie wissen, daß CKOUT den Inhalt des Akkus verändert, in unserem Fall das High-Byte des Zeigers auf den auszugebenden String. Bevor die Ausgabe auf den Befehlskanal gelegt wird, wird daher der Akkuinhalt auf den Stapel gerettet und nach Rückkehr von CKOUT wieder geholt.

Durch das Umleiten der Ausgabe gibt STROUT den betreffenden String auf dem Befehlskanal aus. Mit CLRCH werden zum Abschluß wieder die Standardgeräte für die Ein-/Ausgabe gesetzt.

Nach der Eingabe des MSE-Listing 5 kann die Routine mit SYS 49152 aufgerufen werden. Sie können nun beliebige Disketten-ID's jederzeit ohne Neuformatierung und entsprechenden Datenverlust ändern.

Beachten Sie die folgenden Punkte, wenn Sie Programme erstellen, die Direktzugriffsbefehle verwenden:

1. Die Direktzugriffsbefehle werden immer (!) als String über den Befehlskanal (Sekundäradresse 15) an die Floppystation gesendet. Auch die zugehörigen Parameter (log. Filenummer, Laufwerk (0), Spur, Sektor, Byte) müssen als String übergeben und durch Kommata voneinander getrennt werden.

2. Wenn Sie einzelne Bytes eines Sektors verändern wollen (zum Beispiel die für den Diskettennamen verwendeten Bytes), müssen (!) Sie diesen Block unbedingt zuvor einlesen, da alle anderen Daten des Blocks unverändert auf die Diskette zurückgeschrieben werden sollen. Wenn Sie Directory-Manipulationen vornehmen, ohne diese Bedingung zu beachten, können Sie problemlos das Directory ruinieren und dadurch jeden weiteren Zugriff auf die Diskette – fast – unmöglich machen.

3. Denken Sie bitte daran, daß es beim Schreiben von Daten nicht genügt, den Pufferpointer auf die gewünschte Position zu setzen und die Daten über den Datenkanal an die Floppy zu senden. Ohne Abschluß der Schreibzugriffe mit dem Befehl »U2« befinden sich die Daten zwar im Floppy-Puffer, der Pufferinhalt wurde jedoch noch nicht auf die Diskette selbst geschrieben.

Dieses Problem kennen Sie bestimmt von der Arbeit mit sequentiellen Dateien in Basic, wo Sie ebenfalls Daten verlieren, wenn die Datei nach Schreibzugriffen nicht ordnungsgemäß geschlossen wird.

NAME	FUNKTION	PARAMETER HIN	PARAMETER ZURUECK	ADRESSE
SETFLS	FILEPARAMETER SETZEN	AKKU=LF; X=GA; Y=SA		$FFBA
SETNAM	DATEINAME ÜBERGEBEN	AKKU=LÄNGE; X/Y=POINT. AUF NAME		$FFBD
OPEN	LOG. DATEI ÖFFNEN	VORBEREITUNG: SETFLS, SETNAM	FEHLER: SEC, FEHLER NR.IM AKKU	$FFC0
CLOSE	LOG. DATEI SCHLIESSEN	AKKU=LOG. FILENUMMER	FEHLER: SEC, FEHLERNR. IM AKKU	$FFC3
CLALL	SCHLIESST ALLE FILES			$FFE7
CHKIN	EINGABE VON LOG. DATEI	X=LOG. FILENUMMER	FEHLER: SEC, FEHLERNR. IM AKKU	$FFC6
CKOUT	AUSGABE AUF LOG. DATEI	X=LOG. FILENUMMER	FEHLER: SEC, FEHLERNR. IM AKKU	$FFC9
READST	I/O-STATUS ABFRAGEN		STATUS IM AKKU (BIT 6=DATEIENDE)	$FFB7
BASIN	ZEICHEN EINLESEN		ZEICHEN IM AKKU	$FFCF
BSOUT	ZEICHEN AUSGEBEN	ZEICHEN IM AKKU		$FFD2
LISTEN	'LISTEN' SENDEN	AKKU=GERÄTEADRESSE		$FFB1
UNLISTEN	'UNLISTEN' SENDEN			$FFAE
CHKKOM	KOMMA LESEN			$AEFD
GETBYT	EIN-BYTE-WERT LESEN		X=ÜBERGEBENES BYTE	$B79E
STRPOS	VARIABLENADRESSE HOLEN		$47/$48=POINT. AUF LÄNGENDESCR.	$B08B
STRRES	STRINGPLATZ RESERVIEREN	AKKU=STRINGLÄNGE		$B4F4

Tabelle. Übersicht über alle im Artikel erwähnten Betriebssystem- und Interpreterroutinen

Mein Wissen über die Dateiverwaltung in Maschinensprache ist nun erschöpft, und ich kann Ihnen nur viel Erfolg bei der Programmierung wünschen. Wie Sie sehen, gibt es keinerlei prinzipiellen Unterschiede zur Dateiverwaltung in Basic. Auch Direktzugriffsdateien können problemlos verwaltet werden, wenn man weiß, wie Befehlsstrings an die Floppystation übermittelt werden können.

Den Schluß dieses Artikels bildet eine Tabelle, in der alle verwendeten Betriebssystemroutinen aufgelistet sind. Wenn Sie einzelne der vorgestellten Demoprogramme weiterverwenden wollen, halten Sie sich bitte zur Eingabe an die MSE-Listings.

Auch sei an dieser Stelle auf unser Sonderheft 8/85 mit dem Thema »Assembler« verwiesen, in dem Sie auch den Assembler »Hypra-Ass« finden. (Said Baloui/tr)

Arbeiten mit dBase II

Das Datenbanksystem dBase II von Ashton-Tate hat sich nicht nur auf IBM-PCs und Kompatiblen einen Namen gemacht, es gehört auch zur professionellsten Software, die derzeit für den C128 verfügbar ist. In diesem Beitrag geben wir allen, die noch nicht mit dBase II gearbeitet haben, einen kurzen Einstieg und den Anwendern kompakte Nachschlagetabellen zur täglichen Arbeit an die Hand.

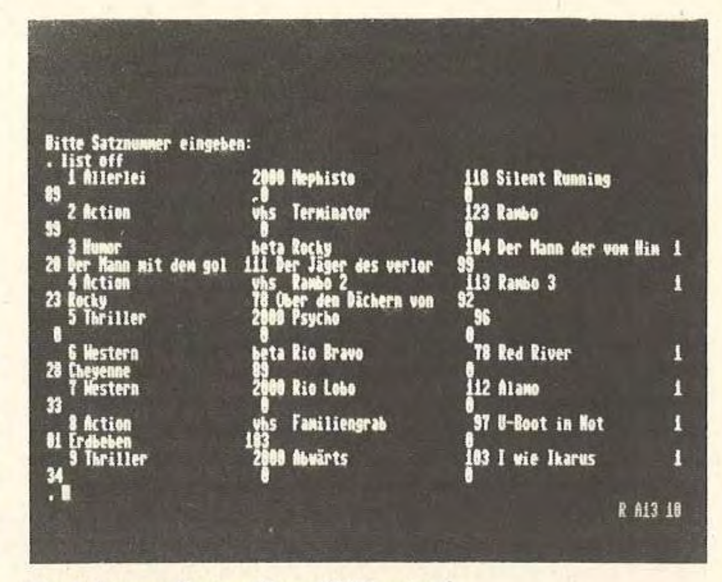


Bild 1. Eine gelistete dBase II-Videodatei

Wir wollen uns zuerst schrittweise vom Starten des Programmes dBase II über die Erstellung einer Datei bis zur Dateneingabe vorarbeiten. Bei den »Vorarbeiten« werden einige Möglichkeiten, die uns das Betriebssystem CP/M zur Verfügung stellt, um schneller und effektiver mit dem Datenbanksystem umgehen zu können, zur Sprache gebracht. Diese »Möglichkeiten« sind zwar für den Programmablauf nicht erforderlich, bieten aber für manchen eine wichtige Anregung.

Erstellen einer dBase II-Arbeitsdiskette

Stellen Sie sich zuerst, wie im Handbuch ausführlich beschrieben, eine Arbeitskopie der dBase II-Disketten her. Sofern Sie dabei die Floppy 1571 verwenden, ergibt sich schon nach dem Formatieren im 1571-Modus (»C128 double sided« im Programm »Format«) ein Geschwindigkeits- und Speicherplatzvorteil gegenüber der Originaldiskette. Dies liegt daran, daß Software für den C128 üblicherweise im 1541-Modus aufgezeichnet wird, um die Möglichkeit zu bieten, sowohl mit der Floppy 1541 als auch den Folgemodellen 1570 und 1571 die Disketten zu verwenden. Da die Floppy 1571 die Disketten beidseitig verwendet und zudem ihr eigenes Format weit schneller einlesen kann, sollte man bei der Formatierung nur den »double-sided«-Modus verwenden. Sollten Sie sich dennoch für den 1541-Modus entscheiden, dürfen auf der vermeintlichen »Rückseite« der Diskette keine wichtigen Programme oder Datenbestände aufbewahrt werden, zumindest nicht ohne Schreibschutz auf dieser Seite. Das Programm »Format« formatiert nämlich auch gegen Ihren Willen im einseitigen Modus die zweite Seite mit. Soviel zum Formatieren. Da auf der dBase II-Arbeitsdiskette nach dem Kopieren die Programme »CPM+.SYS« und »CCP.COM« enthalten sein sollten (um nicht das Betriebssystem von einer anderen Diskette einlesen zu müssen), besteht zuerst die Möglichkeit, die Bildschirmfarben im 40- und 80-Zeichenmodus beliebig den eigenen Bedürfnissen und Wünschen anzupassen. Starten Sie hierfür das Programm »Keyfig«, wählen Sie die Option »Definitions on the CP/M boot disk« an und legen die dBase II-Arbeitsdiskette (auf der das System dann enthalten sein sollte) ein. Wählen Sie nun das Ändern der »logischen« Farben und danach die Art Ihres Bildschirmes (40- oder 80-Zeichen-Darstellung) an. Wenn Sie nun als logische Farbe »a« für Schwarz (=Hintergrund) drücken und anstelle deren beispielsweise »g« für Dunkelblau wünschen, müssen Sie nur noch bei »Select physical color to assign« »g« drücken, fertig. Mit der Schriftfarbe (»e«) können Sie dann auf gleiche Weise verfahren. <RETURN> beendet die Farbzuweisung. Nachdem Sie »done logical/physical colors« anwählten, erscheint die Frage, wohin die neuen Werte gespeichert werden sollen. Da wir die farblichen Änderungen dauerhaft nach jedem »Booten« des CP/M-Systems zur Verfügung haben möchten, speichern wir also auf der »CP/M boot disk«. Davon ausgehend, daß Sie zu diesem Zeitpunkt keine weiteren Änderungen beabsichtigen, können Sie nun die Schlußabfrage mit »N« beantworten, das System neu starten und Ihre Farbkombination begutachten.

Auf die gleiche Weise läßt sich mit dem Programm »Keyfig« auch die Tastatur (also nicht nur die Funktionstasten) mit anderen als den dort gewohnten Tasten belegen, lassen sich sogar Zeichenketten, Farbcodes und spezielle Funktionen programmieren! Dies hört sich vielleicht schwierig an, ist es jedoch nicht. Es nimmt allenfalls etwas Zeit in Anspruch, sich selbst klar darüber zu werden, wo man nun am günstigsten was unterbringt. Den Funktionstasten könnten Sie beispielsweise Strings wie »display structure« oder »display files like `*.*`« zuordnen. Geben Sie am Ende des Textes noch <CTRL+M> ein, so wird ein RETURN (CHR$(13) angehängt und der Befehl gleich ausgeführt.

Die letzte hier vorgestellte Möglichkeit, sich selbst den Umgang mit CP/M und dBase II zu erleichtern, besteht darin, eine Stapelverarbeitungsdatei einzurichten. In diese Datei können Sie dann Programmnamen eintragen, die ausgeführt werden sollen und können diesen Programmen sogar noch Parameter übergeben. Nennt man diese Datei dann noch »Profile.sub«, so wird deren Inhalt bearbeitet, bevor überhaupt die Bereitschaftsmeldung »A« erscheint. Befindet sich die Datei »Profile.sub« mit folgendem Inhalt auf der Diskette

```
setup
<g
<u
dbase
```

so wird zuerst »Setup.com« geladen, der Wert »g« für »German« (deutsche Tastaturbelegung) und »u« für Druckerbetrieb über den User-Port übergeben (die Meldung »Program input ignored« können Sie in diesem Zusammenhang ignorieren) und schließlich »dBase II« ausgeführt. Die spitze Klammer nach links signalisiert, daß die folgenden Zeichen als Pseudo-Tastatureingaben zu verstehen sind. Es lassen sich auch mehrere Zeichen hintereinander auf diese Weise verwenden. Haben Sie unter dBase II schon ein Programm geschrieben, kann man dies gleich berücksichtigen, etwa »dbase video«. Ist dies alles installiert, müssen Sie lediglich noch die Arbeitsdiskette einlegen und den C128 einschalten, alles weitere wird »selbständig« von »Profile.sub« veranlaßt. Sofern Sie keine eigene Anwendung in Form eines Pro-

gramms unter dBase II unmittelbar starten möchten, dafür aber auf das Datum verzichten wollen, müssen Sie an unser obiges Beispiel nur noch eine einzelne spitze Klammer anfügen. Ohne folgende Zeichen wird dies als <RETURN> interpretiert, was bedeutet, daß die Eingabe des Datums übersprungen wird. Auch die Frage, wie die Datei »Profile.sub« zu erstellen ist, soll nicht unbeantwortet bleiben. Sie benötigen hierfür den Editor »Ed«, der hier mit »Ed profile.sub« aufgerufen wird. Im Commodore-Handbuch zu CP/M auf dem C128 finden Sie unter den Kapiteln 7 - 55 bis 7 - 58 eine Aufstellung der Editorkommandos (die sehr gewöhnungsbedürftig sind). Hier sollen nur in Kürze die uns direkt betreffenden Anweisungen beschrieben werden. Nachdem sich der Editor mit dem »*« meldet, geben Sie »I« für Insert (Einfügen) ein; da (noch) kein Inhalt vorhanden ist, wird das Einfügen von Text am Anfang durchgeführt. Es erscheint »1:« auf dem Bildschirm, Sie sind also in der ersten Zeile. Nachdem Sie in einem »manuellen Durchgang« festgestellt haben, welche Programme mit welchen Parametern Sie für Ihre Zwecke benötigen und dies am besten gleich notiert haben, tragen Sie nun Zeile für Zeile Ihre Anweisungen ein. Mit Ausnahme der letzten Eingabezeile verwenden Sie immer <RETURN>, um die Zeile abzuschließen. In besagter letzter Zeile drücken Sie <CTRL+Z>, es erscheint dann wieder das Sternchen (»*«), und Sie können mit <E> und <RETURN> die neu erstellte Datei auf Ihre Arbeitsdiskette schreiben. Nach diesen (nicht dringend erforderlichen) Vorbereitungen, deren Prinzip zudem bei allen Anwendungen unter CP/M verwendet werden kann, steht Ihnen das dBase II-System mit allen Einstellungen Ihrer Wahl zur Verfügung, nachdem Sie lediglich die Diskette eingelegt und den C128 eingeschaltet haben!

dBase II-Kompaktkurs

Da eine ausführliche Beschreibung aller Befehle und Funktionen von dBase II selbst den Rahmen dieses Sonderheftes sprengen würde, wurde ein Kompromiß zwischen Kurs und Nachschlagewerk gewählt. Im folgenden finden Sie ein alphabethisches Verzeichnis aller dBase II-Befehle mit Kurzbeschreibung (Tabelle 1), ein Verzeichnis aller Funktionen (Tabelle 2) und eine Übersicht über alle wichtigen Begriffe rund um dBase II (Tabelle 3). Weiterhin werden alle Cursortasten in den verschiedenen Modi aufgeführt (Tabelle 4), die dBase II-Befehle mit Beispielen erklärt (Tabelle 5), die Fehlermeldungen sowie deren Ursache/Beseitigung zusammengefaßt (Tabelle 6) und schließlich noch wichtige Systemdaten (Tabelle 7) und Literaturhinweise (Info) gegeben.

Sie erhalten jederzeit im Programm Hilfestellung zum dBase II-System, indem Sie »HELP« und den Namen des dBase II-Kommandos oder »HELP« und eines der Schlüsselwörter

- ANWENDERPROGRAMME	-BEGRIFFE
- -HELP	-NEU
- KENNDATEN	-BEISPIELE
- FUNKTIONEN	-DBASE
- BILDSCHIRM	-MODI
- -FEHLER	-INSTALL

eingeben. Das Schlüsselwort muß dabei vollständig eingegeben werden. Sofern das Schlüsselwort nicht gefunden werden kann (etwa aufgrund eines Eingabefehlers oder weil die HELP-Datei gelöscht wurde), erscheint die Systemmeldung »Keine HELP-Texte gefunden«. Umfaßt ein HELP-Text mehrere Bildschirmseiten, so wird am Ende jeder Seite »System wartet« ausgegeben. Sofern Sie auf der dBase II-Arbeitsdiskette die HELP-Datei entbehren können (kopieren Sie in diesem Fall »DBASEMSG.TXT« am besten auf eine ungenutzte Diskette und löschen die Datei mit »erase« auf der Arbeitsdiskette), gewinnen Sie wertvollen Speicherplatz (etwa 60 KByte), was besonders bei Verwendung nur eines Laufwerkes wichtig werden kann.

Die wichtigsten dBase II-Dateien befinden sich auf der ersten Diskette. Dies sind:

DBASE.COM	- Hauptprogramm
DBASEOVR.COM	- Übersichten und System-Mitteilungen
DBASEMSG.TXT	- HELP-Datei (nur erforderlich, wenn Sie »HELP« nutzen)
INSTALL.COM	- Programm zur Anpassung an das Terminal (die Version für den C128 ist bereits angepaßt).

Wie schon erwähnt, ist das Programm »INSTALL.COM« sowie die Textdatei »DBASEMSG.TXT« entbehrlich; Sie sollten beide aber vorsichtshalber auf eine eigene Diskette kopieren, für den Fall, daß die Systemdisketten beschädigt werden.

(O.Trottno/bj)

Tabelle 1: Alle dBase II-Befehle auf einen Blick

?	- zeigt einen Ausdruck, eine Variable oder ein Feld an
??	- zeigt eine Liste von Ausdrücken an, ohne vorher einen Zeilenvorschub auszuführen
@	- gibt formatierte Daten auf Bildschirm oder Drucker aus
ACCEPT	- Eingabe einer Zeichenkette in eine bestimmte temporäre Variable
APPEND	- fügt Informationen von einer anderen dBase II-Datenbank oder -Datei an die benutzte Datei an
BROWSE	- Bildschirm-Darstellung und -Bearbeitung der Datenbank
CANCEL	- beendet die Ausführung einer Programmdatei
CHANGE	- feldweise Bearbeitung von Datenbanken
CLEAR	- schließt alle eröffneten Datenbanken und löscht alle erzeugten temporären Variablen
CONTINUE	- setzt die Suchaktion nach einem LOCATE-Befehl fort
COPY	- kopiert Daten aus einer Datei in eine andere Datei
COUNT	- zählt die Datensätze in einer Datei, die eine angegebene Bedingung erfüllen
CREATE	- erzeugt eine neue Datenbank-Datei
DELETE	- löscht eine Datei oder markiert einen Satz zum Löschen
DISPLAY	- zeigt Dateien, Datensätze, Strukturen, temporäre Variable oder den Status an
DO	- führt Programmdateien oder strukturierte Schleifen aus
EDIT	- ruft die Bearbeitung von Daten in einer Datenbank auf
EJECT	- erzeugt Seitenvorschub auf dem Drucker
ELSE	- alternativer Weg bei der Befehlsausführung innerhalb IF
ENDCASE	- beendet einen CASE-Befehl
ENDDO	- beendet einen DO WHILE-Befehl
ENDIF	- beendet einen IF-Befehl
ENDTEXT	- beendet einen TEXT-Befehl
ERASE	- löscht den Bildschirm
FIND	- sucht einen Datensatz in einer indizierten Datei
GO/GO TO	- geht zu einer bestimmten Position in einer Datei
HELP	- gibt dem Benutzer Hilfestellung auf dem Bildschirm

IF	– erlaubt eine bedingte Befehlsausführung
INDEX	– erzeugt eine Index-Datei
INPUT	– erlaubt Eingabe von Ausdrücken in temporäre Variable
INSERT	– einfügen neuer Datensätze
JOIN	– erzeugt gemeinsame Ausgabe aus zwei Datenbank-Dateien
LIST	– listet Dateien, Datensätze, Strukturen, temporäre Variable und den Status
LOCATE	– findet einen Datensatz, der eine Bedingung erfüllt
LOOP	– springt zum Anfang eines DO WHILE-Befehls
MODIFY	– erzeugt und/oder bearbeitet eine Programm-Datei oder ändert die Struktur in einer Datenbank-Datei
NOTE oder	– Kommentar-Einleitung in einem Programm
PACK	– eliminiert Datensätze, die zum Löschen markiert sind
QUIT	– verläßt dBase II und geht zum Betriebssystem zurück
READ	– erlaubt Datei-Bearbeitung mit formatiertem Bildschirm, nimmt die Daten aus SAY- und GET-Anweisungen entgegen
RECALL	– löscht die Markierungen zum Löschen von Datensätzen
REINDEX	– aktualisiert die vorhandene Index-Datei
RELEASE	– eliminiert unerwünschte temporäre Variablen und macht Speicherbereiche frei
REMARK	– erlaubt die Anzeige beliebiger Zeichen
RENAME	– gibt einer Datei einen neuen Namen
REPLACE	– ändert Informationen in einem Datensatz oder in einer ganzen Datenbank-Datei Feld für Feld
REPORT	– erzeugt einen Bericht
RESET	– startet das Betriebssystem nach dem Einlegen einer neuen Diskette
RESTORE	– reaktiviert temporäre Variable aus einer Datei, gegebenenfalls in Ergänzung zu bereits vorhandenen Variablen
RETURN	– beendet den Lauf einer Programm-Datei
SAVE	– speichert die temporären Variablen auf der Diskette
SELECT	– schaltet zwischen primärer und sekundärer Datenbank um
SET	– setzt dBase II-Kontrollparameter
SKIP	– springt vorwärts und rückwärts in der Datenbank
SORT	– erzeugt eine Datei, die nach einem Schlüsselfeld sortiert ist
STORE	– erzeugt temporäre Variablen
SUM	– berechnet die Gesamtsummen der Felder in einer Datenbank
TEXT	– gibt Textblöcke von einer Programm-Datei aus
TOTAL	– erzeugt zusammengefaßte Kopien einer Datenbank, bestehend aus Daten bestimmter Felder oder Datensätze
UPDATE	– aktualisiert eine Datenbank im Stapelverarbeitungsbetrieb
USE	– eröffnet eine Datenbank für nachfolgende Bearbeitung, bis der nächste USE-Befehl erscheint
WAIT	– unterbricht die Programmausführung, bis eine Eingabe vom Benutzer erfolgt ist

Tabelle 2. Alle dBase II-Funktionen

@	– @(<Zeichenkette1>,<Zeichenkette2>): Die »AT«-Funktion ergibt eine ganze Zahl für die Position in Zeichenkette2, wo Zeichenkette1 beginnt.
*	– Löschmarkierung; führt zum Überlesen des Satzes bei COPY oder ähnlichen Operationen.
#	– Satznummer-Funktion: Sie zeigt einen ganzzahligen Wert an, der der laufenden Satznummer entspricht.
!	– !(<Zeichenkette>): Großbuchstaben-Funktion. Sie wandelt die Zeichen in Großbuchstaben um.
$	– $(<Zeichenkette>, <Start>,<Länge>): Die Unterketten-Funktion erzeugt eine Zeichenkette aus dem angegebenen Teil einer anderen Zeichenkette.
CHR	– CHR(<numerischer Ausdruck>) ergibt das ASCII-Zeichen entsprechend dem numerischen Ausdruck. Beispielsweise sendet ».? CHR(7)« das Escape-Signal an den Drucker
DATE()	– gibt das Systemdatum in der Form xx/xx/xx aus.
EOF	– Datei-Ende-Funktion: Sie ist wahr, wenn versucht worden ist, über den letzten Datensatz der Datei hinauszugehen.
FILE	– FILE(<Datei>): Die Existenz-Funktion ist wahr, falls <Datei> auf dem angegebenen Laufwerk vorhanden ist.
INT	– INT (<numerischer Ausdruck>): Die Ganzzahl-Funktion schneidet alle Stellen rechts vom Dezimalpunkt ab, um eine ganze Zahl zu bilden.
LEN	– LEN(<Zeichenkette>): Die Längen-Funktion zeigt die Anzahl der Zeichen in der Zeichenkette an, beispielsweise ».?? LEN('KLAUS')« ergibt 5
RANK	– RANK(<Zeichenkette>) gibt den numerischen ASCII-Code des ersten Zeichens in der Zeichenkette wieder.
STR	– STR (<numerischer Ausdruck>, <Weite:,<Dezimalstelle>:). Die String-Funktion wandelt einen numerischen Ausdruck in eine Zeichenkette um.
TEST	– TEST(<Ausdruck>) bildet zusammen mit »?« und »IF« die Test-Funktion. Sie prüft, ob ein Ausdruck gültig und passend ist. Ein gültiger Ausdruck liefert einen Wert ungleich 0; ein ungültiger den Wert 0.
TRIM	– TRIM(<Zeichenkette>): Die Trim-Funktion entfernt nachlaufende Leerstellen der Zeichenkette.
TYPE	– TYPE(<Ausdruck>) erzeugt eine einstellige Zeichenkette, die »C«,»N«,»L« oder »U« enthält, je nachdem, ob der Ausdruck vom Typ Character (Zeichen), Numeric (numerisch), Logical (logisch) oder Undefined (unbestimmt) ist.
VAL	– VAL(<Zeichenkette>): Die Wert-Funktion wandelt eine aus Zahlen bestehende Zeichenkette in einen numerischen Wert um.

Tabelle 3. Begriffe rund um dBase II

<Kommando> oder <Aussage>	– alle gültigen dBase II-Kommandos (Befehle) oder -Funktionen.
<Zeichenkette>	– Sie müssen in den meisten Fällen mit einfachen (' '), doppelten Anführungszeichen (" ") oder Klammern (: :) begrenzt sein.
<Zeichenkettenausdruck>	– Ausdruck, dessen Inhalt vom Zeichentyp her festgelegt ist.
<Begrenzer>	– jedes nicht alphanumerische Zeichen, das benutzt wird, um Daten zu kennzeichnen, zum Beispiel der Apostroph (' '), die Anführungszeichen (" ") oder Klammern.
<Ausdruck>	– ein Zeichen oder eine Zeichengruppe, deren Wert von dBase II bestimmt werden kann. <Ausdruck> ist abhängig vom Datentyp und wird mit »C«, »N«, oder »L« bezeichnet.
<Ausdruckliste>	– eine Liste durch Komma getrennter Ausdrücke.
<Feld>	– Name eines Datenfeldes innerhalb eines Datensatzes.
<Feldliste>	– eine Liste mit Feldnamen, getrennt durch Kommatas.
<Datei>	– Name der Datei, die Sie erzeugen oder bearbeiten wollen.
<Indexdatei>	– Name der Indexdatei.
<Schlüssel>	– Felder (auch kombiniert) zum Indizieren von Datenbankdateien
<tempVar>	– Der Name einer temporären Variablen.
<tempVarliste>	– Liste mit temporären Variablen, getrennt durch Kommatas.
<n>	– eine Zahl, die dBase II als Buchstaben betrachtet.
<num Ausdr>	– Ausdruck mit numerischem Inhalt.
<Bereich>	– das Kommando legt einen Bereich in einer Datenbankdatei fest, der bei Ausführung eines Kommandos berücksichtigt wird. <Bereich> kann etwa sein: ALL(e) Sätze der Datei; NEXT <n> Sätze der Datei und RECORD <n>. Der Standardwert ist vom jeweiligen Befehl abhängig.
<Muster>	– erlaubt eine Stapelverarbeitung von Dateien desselben Typs und/oder mit passenden Zeichen im Dateinamen, etwa bei Benutzung von »wild cards« und »?«. Es dient weiterhin zur Auswahl einer Gruppe von temporären Variablen.
<Variable>	– Name eines Datenfeldes oder einer <tempVar>.
: :	– schließt optional zu verwendende Angabe ein (statt eckiger Klammern).

dBase II für den Commodore 128 PC. Im Lieferumfang des Programms sind enthalten:

- zwei Disketten mit der auf den C128 angepaßten Version von dBase II, Druckerinstallationsprogramm, Help-Datei und Beispiele
- ein über 460 Seiten umfassendes Handbuch (Hardcover) mit Einführungsteil, Befehlsbeschreibung, Beispielen und Stichwortverzeichnis.

Bezugsquelle: Markt&Technik Verlag AG, Hans-Pinsel-Straße 2, 8013 Haar bei München. Preis: 199 Mark.

dBase II ist ein eingetragenes Warenzeichen von Ashton-Tate, Culver City, USA
CP/M ist ein eingetragenes Warenzeichen der Digital Research Inc., USA

Tabelle 4. Alle Cursorbewegungen auf einen Blick

Bildschirm-Modus	
<CTRL+X>	bewegt Cursor abwärts zum nächsten Feld (wie <CTRL+F>)
<CTRL+E>	bewegt Cursor aufwärts zum vorherigen Feld (wie <CTRL+A>)
<CTRL+D>	bewegt Cursor ein Zeichen nach rechts (vor-wärts)
<CTRL+S>	bewegt Cursor ein Zeichen nach links (rück-wärts)
<CTRL+G>	löscht das Zeichen unter dem Cursor
<DEL>	löscht das Zeichen links vom Cursor
<CTRL+Y>	löscht das laufende Feld rechts vom Cursor
<CTRL+V>	schaltet um zwischen Überschreib- und INSERT-Modus
<CTRL+W>	speichert Änderungen und geht zurück zum "."-Prompt
Editier-Modus	
<CTRL+U>	schaltet die Markierung DELETE des Satzes ein beziehungsweise aus
<CTRL+C>	schreibt laufenden Satz auf Diskette, geht zum nächsten
<CTRL+R>	schreibt laufenden Satz auf Diskette, geht zum vorherigen
<CTRL+Q>	ignoriert Änderungen im Satz und geht zurück zum "."-Prompt
<CTRL+W>	speichert alle Änderungen und geht zurück zum "."-Prompt
Browse-Modus	
<CTRL+B>	verschiebt das Bildschirm-Fenster ein Feld nach rechts
<CTRL+Z>	verschiebt das Bildschirm-Fenster ein Feld nach links
Modify-Modus	
<CTRL+T>	löscht die laufende Zeile, zieht alle unteren Zeilen hoch
<CTRL+N>	Einfügen einer neuen Zeile an der Cursorposition
<CTRL+C>	schiebt Bildschirm eine halbe Seite abwärts
<CTRL+W>	speichert alle Änderungen und geht zurück zum "."-Prompt
<CTRL+Q>	ignoriert alle Änderungen und geht zurück
Append+Modus	
<enter>	beendet APPEND, wenn der Cursor in der ersten Position des ersten Feldes ist
<CTRL+W>	speichert den Datensatz und geht zum nächsten Satz
<CTRL+Q>	ignoriert den laufenden Satz, geht zurück zum "."-Prompt
Sonstige wichtige Control-Tasten	
<CTRL+P>	schaltet Ihren Drucker an und aus
<CTRL+R>	wiederholt das zuletzt ausgeführte Kommando
<CTRL+X>	löscht die Kommandozeile, ohne ein Kommando auszuführen
<CTRL+H>	Rücklauf
<CTRL+M>	erzeugt einen Wagenrücklauf
<CTRL+S>	startet/stoppt CPU-Operationen

Tabelle 5. dBase II-Befehlsbeschreibung und Beispiele

? – erarbeitet und zeigt den Wert eines Ausdrucks an. Kann beispielsweise in Programmdateien und im Befehlsmodus verwendet werden, um das Ergebnis eine Zeile tiefer anzuzeigen. Auf einige Besonderheiten (die vom ähnlichen Basic-Befehl PRINT abweichen) soll hier etwas näher eingegangen werden. Die Division, etwa ».?12/7« ergibt 1. Tauchen jedoch in Divident oder Divisor Nachkommastellen auf (also auch .000), so wird im Ergebnis die höchste Anzahl an Nachkommastellen beider Werte ermittelt und das Ergebnis auf gleiche Anzahl an Nachkommastellen berechnet und ausgegeben. Aus ».? 12.0/7.000« resultieren entsprechend 1.714, also drei Nachkommastellen.

Anders verhält es sich jedoch bei der Multiplikation. Hier wird die Anzahl der Nachkommastellen addiert, ».? 4.0*4.0 *5.00« ergibt demnach 80.0000. Addition und Subtraktion verhalten sich wie die Division. Weiterhin lassen sich dadurch Speichervariable, die Inhalte von Feldern oder auch komplette Dateien lesen.

?? – wie ?, zeigt allerdings das Ergebnis in derselben Zeile an.

@ – Ausgabe (formatierter) Daten auf Bildschirm oder Drucker beginnend bei x,y-Koordinate (x=Zeile, y=Spalte).
Syntax:
@ <Koordinaten> SAY <Ausdr> USING '<picture>'
GET <Variable> PICTURE '<picture>'
Beispiele:
@3,23 SAY WERT 1.06 USING '$$$,$$$.99'
@14,23 SAY "TEL-NR EINGEBEN" GET TEL PICTURE '(###) ###-####
@LINE+2,45 SAY TOTAL USING '99999.99'

ACCEPT – Eingabe einer Zeichenkette in eine bestimmte temporäre Variable, die dann vom Typ »C« ist. Als Meldung erscheint die Zeichenkette mit nachfolgendem Doppelpunkt. Die Eingabe wird in <tempVar> abgelegt.
Syntax:
ACCEPT "<Zeichenkette>" TO <tempVar>
Beispiele:
. ACCEPT "Wie ist Ihr Name?" TO NAME
Wie ist Ihr Name?: (Antwort wird in NAME gespeichert)
. ? NAME
(Antwort wird auf dem Bildschirm angezeigt)

APPEND <Datei> DELIMITED – APPEND FROM <Datei> FOR <Ausdr> oder APPEND FROM <Datei>
– fügt Datensätze von einer Datenbank-Datei oder einer Format-Datei an die gerade benutzte Datenbank-Datei an. Datensätze, die in der Ursprungsdatei zum Löschen markiert sind, werden nicht übernommen.
zum Beispiel APPEND FROM ADRESSLISTE FOR NAME='N'
APPEND FROM TEST.TXT DELIMITED

APPEND BLANK – fügt einen leeren Datensatz an die Datei an.

APPEND – fügt neue Datensätze an die benutzte Datenbank-Datei an. Sofern ein Index benutzt wird, wird die Index-Datei ebenfalls aktualisiert.

BROWSE FIELDS <Feldliste> – Ausgabe der Felder auf Bildschirm (Voreinstellung: alle) und Bearbeitungsmöglichkeit dort mit freier Cursorpositionierung.

CANCEL – beendet in einer Programmdatei die Dateibearbeitung und geht zurück zum ».«-Prompt.
Ausschnitt aus Programmdatei:
ACCEPT "Was soll diese Maschine machen?" TO NEXT
IF NEXT = 'Q'
CANCEL
ENDIF

CHANGE – erlaubt eine feldweise Bearbeitung von Datenbanken. Drücken der <ESC>-Taste beendet CHANGE-Modus.
Syntax: CHANGE <Bereich> FIELD <Liste> FOR <Ausdr>
Beispiel:
CHANGE ALL FIELD PLZ FOR PLZ = '8000'
RECORD: 00001
PLZ: 8000
CHANGE?
Zu änderndes Zeichen eingeben und <RETURN> drücken. Neue Daten mit dem TO-Prompt einfügen oder <RETURN> drücken, um den nächsten gewünschten Datensatz zu erhalten.

CLEAR – schließt alle eröffneten Datenbanken und löscht alle erzeugten temporären Variablen und Arbeitsbereiche.

CLEAR GETS – weist dBase II an, alle noch aktiven GET-Angaben zu eliminieren, ohne den Bildschirm zu löschen.

COPY – kopiert Daten aus der benutzten Datenbank oder deren Struktur in eine andere Datei. COPY erzeugt die neue Datei, falls sie noch nicht existiert, kann aber eine bereits vorhandene Datei mit gleichem Namen zerstören. Zum Löschen markierte Datensätze werden nicht kopiert.
Das Kommando hat drei Formen:
COPY TO <Datei> <Bereich> FIELD <Liste> FOR <Ausdr>
COPY TO <Datei> SDF DELIMITED WITH <Begrenzer> FOR <Ausdr>
COPY TO <Datei> STRUCTURE FIELD <Liste>

COPY TO <Datei> – STRUCTURE EXTENDED – spezielle Form von COPY. Erzeugt Datenbank, deren Datensätze die Struktur der benutzten Datei beschreiben.

COUNT – zählt die Anzahl der Datensätze in der benutzten Datei, die eine angegebene Bedingung erfüllen. Standardwert ist: COUNT alle Datensätze.
Syntax:
COUNT <Bereich> FOR <Ausdr> TO <tempVar>
Beispiel:
COUNT NEXT 25 FOR TEMPERATUR > 'B' TO LISTE

CREATE <Dateiname> – erzeugt eine neue Datenbank. Der Benutzer wird nach der Dateistruktur gefragt.

CREATE <Neudatei> FROM <Altdatei> EXTENDED – erzeugt eine Datenbank, deren Struktur bestimmt wird durch die Daten, die in den Datensätzen der alten Datei enthalten sind (siehe auch COPY STRUCTURE EXTENDED).

DELETE FILE <Datei> – löscht eine bestimmte Datei.

DELETE <Bereich> FOR <Ausdr> – markiert Sätze als gelöscht. Datensätze, die mit " " als gelöscht markiert sind, werden erst entfernt, wenn das PACK-Kommando gegeben wird. Datensätze können auch durch die Satznummer angegeben werden.
Beispiele:
DELETE ALL FOR FIRMA = 'ZMB'
DELETE RECORD 15

DISPLAY FILES ON <Laufwerk> LIKE <Muster>: – zeigt Inhaltsverzeichnis des Diskettenlaufwerkes.
Zum Beispiel:
DISPLAY FILES ON B LIKE .MEM.

DISPLAY <Bereich> <Feldliste> FOR <Ausdrliste> OFF – zeigt die Datensätze der benutzten Datei an.

DISPLAY STRUCTURE – zeigt die Feldnamen, -typen, -längen und Dezimalstellen (Datenstruktur) der benutzten Datei an.

DISPLAY MEMORY – zeigt Namen, Typen und Werte aller temporären Variablen an.

DISPLAY STATUS – listet benutzte Datenbanken, Indizes, Systemangaben und laufende Systemparameter auf.

DO <Datei> – öffnet und führt eine Programmdatei aus.

DO WHILE <Ausdr> – führt in einer Programmdatei eine Gruppe von Anweisungen mehrfach aus (Schleife); wird so lange durchlaufen, wie DO WHILE <Ausdr> wahr ist.
Syntax:
DO WHILE <Ausdr>
Beispiel:
USE ADRESSLISTE
<Befehle>

```
DO WHILE .NOT. EOF
            ? VORNAME
   ? NAME
               ? STRASSE
               ? ORT
   ? TELEFON
   SKIP
ENDDO
```

ENDDO – beendet den Anweisungsblock in einer DO WHILE-Anweisung.

LOOP – veranlaßt die Programmdatei, zur weiteren Bearbeitung zurück zum DO WHILE-Kommando zu springen.
Beispiel:

```
USE ADRESSLISTE
DO WHILE .NOT. EOF
  IF PLZ = '8000'
     SKIP
     LOOP
  ENDIF
  ? NAME
  ? TELEFON
  SKIP
ENDDO
```

DO CASE – wird in Programmdateien benutzt, um die Ausführung fallweise zu steuern. Wahlweise kann die OTHERWISE-Klausel benutzt werden, die dann gilt, wenn kein CASE zutrifft. Der Befehl muß mit ENDCASE beendet werden.
Beispiel:

```
USE ADRESSLISTE
   ACCEPT "WELCHE ALTERNATIVE
   WÜNSCHEN SIE?" TO Wahl
      DO CASE
         CASE Wahl = '1'
              DO Aufkleber
         CASE Wahl = '2'
              DO Anfügen
         CASE Wahl = '3'
              DO Editieren
         OTHERWISE
              Return
      ENDCASE
```

EDIT :<Satznummer>: – erlaubt die gezielte Bearbeitung von Daten durch die Angabe der Satznummer. Fragt nach der Satznummer, falls diese nicht mit dem Befehl angegeben wurde. Wenn die Bearbeitung eines Satzes abgeschlossen ist, wird mit <CTRL+W> wieder das Prompt zur Eingabe der Satznummer erreicht. Um den EDIT-Modus zu beenden, für Satznummer <RETURN> eingeben.

EJECT – veranlaßt beim Drucker einen Seitenvorschub, falls PRINT auf »ON« oder FORMAT auf »PRINT« gesetzt ist. Bei Verwendung von »@ SAY«-Kommandos für formatierte Ausgaben setzt EJECT den Zeilen- und Spaltenzähler auf Null.

ERASE – löscht die Bildschirmanzeige. Im Dialog erscheint der ».«-Prompt in der linken oberen Bildschirmecke.

FIND <Zeichenkette> – sucht einen Datensatz in einer indizierten Datenbank nach dem Wert des Schlüssels <Zeichenkette>.

GO oder GOTO – geht zu einer bestimmten Position (Datensatznummer) in einer Datenbank. Die Datensatznummer kann auch Inhalt einer temporären Variablen sein.
Syntax:
GO oder GOTO :RECORD <n>, <n>, TOP, BOTTOM oder <tempVar>

HELP – gibt dem Anwender Hilfestellung
Syntax:
HELP <Kommandoname> <RETURN> (zum Beispiel HELP CREATE <CR>). Nach Ausgabe der Erläuterung kehrt dBase II zum ».«-

64ER ONLINE

Prompt zurück, so daß Sie ohne Unterbrechung weiterarbeiten können. In einigen Fällen beansprucht die Information mehrere Seiten auf dem Bildschirm; um weiterzublättern, drücken Sie eine beliebige Taste. Falls Sie die HELP-Datei verlassen wollen, bevor Sie alle Angaben gesehen haben, drücken Sie einfach <ESC>.

IF – erlaubt in der Programmdatei die bedingte Ausführung von Befehlen. Die ELSE-Klausel ist optional.

```
Syntax: IF <Ausdr>
Beispiel:
IF STAAT = 'D'
                <Aussage>
   DO INLAND (CMD Datei)
ELSE
                <Aussage>:

   DO AUSLAND (CMD Datei)
ENDIF
```

INDEX ON <Ausdr> TO <Indexdatei> – erzeugt eine Indexdatei für die Datenbank, wobei der Index festgelegt wird durch <Ausdr> oder <Indexschlüssel>.

INPUT "<Zeichenkette>" TO <tempVar> – nimmt Eingaben des Benutzers (numerische und logische Informationen) entgegen und legt sie in temporären Variablen ab.
Beispiel:
INPUT 'Benutzernr. eingeben' TO X
Benutzernummer eingeben: 12 <CR> (Benutzernr., 12, wird in X gespeichert)

INSERT BEFORE BLANK – fügt einen Datensatz in die Datei ein, unmittelbar nach oder – mit BEFORE – vor dem laufenden Datensatz. Zeigt dem Benutzer das Dateneingabeformat für die benutzte Datei an, außer, ein leerer (BLANK) Datensatz wird eingegeben.

JOIN – erzeugt eine neue Datei durch Verbindung der Datensätze aus zwei bestehenden Dateien (primäre und sekundäre Datei). Datensätze werden angefügt, solange wie FOR <Ausdr> wahr ist. Befehl muß im primären Datenbankbereich ausgeführt werden. Standardwert für <Feldliste> ist ALL.
Syntax:
JOIN TO <Datei> FOR <Ausdr> FIELDS <Feldliste>
Beispiel:
. USE NAMES
. SELECT SECONDARY
. USE ADRESSLISTE
. SELECT PRIMARY
. JOIN TO NAMEJ FOR LAST <> S.LAST

LIST FILES ON <Laufwerk> LIKE <Muster> – listet alle Dateien auf dem gewählten Laufwerk. Standardwert ist das gewählte Laufwerk.
Beispiel: LIST FILES ON C LIKE *.FRM

LIST <Bereich> <Feldliste> FOR <Ausdrliste> OFF: – zeigt Datensätze der benutzten Datei an. Standardwert ist ALL (alle) Datensätze.
Beispiel:
LIST NEXT 25 NAME, TELEFON FOR KOSTEN > 100
(zum Beispiel NAME und TELEFON für die nächsten 25, mit mehr als 100 Mark Telefonkosten.

LIST STRUCTURE – zeigt Datenstruktur der benutzten Datei.

LIST MEMORY – listet alle Namen und Werte der festgelegten temporären Variablen auf.

LIST STATUS – listet geöffnete Dateien, benutzte Indizes, Systemdaten und gesetzte Parameter.

LOCATE <Bereich> FOR <Ausdr> – findet ersten Datensatz, für den FOR <Ausdr> als wahr gilt. Verwenden Sie CONTINUE, um den nächsten Datensatz zu finden.
. LOCATE ALL FOR PLZ >= '5000' .AND. PLZ < '4000'
RECORD: 00123
. DISPLAY
. CONTINUE
RECORD: 00232

MODIFY STRUCTURE – erlaubt, die Struktur der benutzten Datenbank zu verändern. Dieser Befehl zerstört alle Daten in der benutzten Datenbank. Um die Struktur zu verändern, ohne Daten zu verlieren, benutzen Sie COPY STRUCTURE, USE und APPEND wie unten beschrieben:
. USE NAMES
. COPY STRUCTURE TO TEMP
. USE TEMP
. MODIFY STRUCTURE – kein Datenverlust, da Datei leer
. APPEND FROM NAMES – bringt Daten in modifizierte Datei
. DELETE FILE NAMES
. USE – TEMP
. RENAME TEMP TO NAMES

MODIFY COMMAND <Datei> – ruft den Texteditor in dBase II auf und zeigt oder erzeugt die gewünschte Datei. Primär zu nutzen zum Erzeugen und Bearbeiten von Programmdateien (.PRG), Textdateien (.TXT) und Formatdateien (.FMT); aber auch als Bildschirm-Wortprozessor einzusetzen.

NOTE oder * – erlaubt das Einfügen von Kommentaren in eine Programmdatei. Kommentare, die nach NOTE oder »*« folgen, werden bei der Programmausführung übergangen.

PACK – eliminiert Datensätze, die zum Löschen markiert sind. Falls eine Indexdatei vorhanden ist, wird sie automatisch aktualisiert.

QUIT TO <Programmdateiliste> – beendet dBASE II und geht zum Betriebssystem zurück. Ruft optional Programmdateien auf.

READ – ermöglicht Bildschirm-Modus für die Eingabe und das Editieren von Variablen. Die Bildschirm-Prompts und -Fenster werden durch SAY- und GET-Kommandos erzeugt.

RECALL <Bereich> FOR <Ausdr> – reaktiviert zum Löschen markierte Datensätze. Standardwert für <Bereich> ist der laufende Datensatz.

REINDEX – aktualisiert diejenigen Indexdateien, die nicht mit USE angeschlossen und daher nicht automatisch geändert wurden.
Beispiel:
. USE ADRESSLISTE INDEX INDEXA
. APPEND
(auszuführende Kommandos)
. SET INDEX TO INDEXB,INDEXC
. REINDEX

RELEASE – eliminiert nicht mehr benötigte temporäre Variable und macht davon belegten Speicherplatz frei. Mit »?« werden einzelne (beliebige) Zeichen zugelassen, mit " " beliebige Zeichenreihen.
Syntax:
RELEASE <tempVarliste> oder ALL
RELEASE ALL LIKE <Muster>
RELEASE ALL EXCEPT <Muster>
Beispiele (mit N, N1, N2, N10, V7 als Variablen):
RELEASE ALL LIKE N? (N10 und V7 bleiben erhalten)
RELEASE ALL LIKE N?? (V7 bleibt erhalten)
RELEASE ALL EXCEPT ?1 (N1 und N10 bleiben erhalten)

REMARK – zur Kommentierung; beliebige Zeichenreihen sind verwendbar. Er wird bei der Ausführung von Programmen überlesen. Beispiel:
. REMARK irgendein Hinweis

RENAME <Datei> TO <neuer Dateiname> – ändert den Namen einer Datei im Verzeichnis des Betriebssystems. Wenn nicht anders vorgegeben, definiert dBase II den Dateityp als DBF.
Beispiel: . RENAME REVIEW.FRM TO REVIEW2.FRM

REPLACE – ändert Daten in bestimmten Datenfeldern oder Dateien. Falls das geänderte Feld gleichzeitig Schlüsselfeld in der Indexdatei ist, erfolgt deren Änderung. Standardwert für <Bereich> ist der laufende Datensatz.
Syntax:
REPLACE <Bereich> <Feld> WITH <Ausdr> ,<Feld2> WITH <Ausdr2>: :FOR <Ausdr>:
Beispiel:
. USE MXPROJ
. REPLACE ALL COST WITH COST 6.1 FOR ITEM = 'ELEC'

REPORT – erzeugt eine Formatdatei für einen Bericht (FRM), um bestimmte Informationen aus einer Datei in einem vom Benutzer festgelegten Format auszugeben. Das Ergebnis erscheint auf dem Bildschirm oder wird ausgedruckt.
Syntax:
REPORT FORM <Formdatei> <Bereich> TO PRINT FOR <Ausdr> PLAIN

RESET <Laufwerk> – nach einem Diskettenwechsel ist dieser Befehl auszuführen. Dadurch können während der Arbeit die dBase II Disketten ausgetauscht werden. Bevor jedoch der Befehl RESET erfolgt, sind unbedingt alle Dateien auf der zu wechselnden Diskette abzumelden.

RESTORE FROM <Datei> ADDITIVE – sucht und aktiviert die Parameter für temporäre Variablen, die vorher mit SAVE gespeichert worden sind. Mit ADDITIVE wird erreicht, daß bereits vorhandene Variablen erhalten bleiben. Fehlt die Angabe, werden alle aktuellen Variablen gelöscht (siehe auch RELEASE).

RETURN – wird in einer Programmdatei benutzt, um zum aufrufenden Programm oder zum dBase II-».«-Prompt zurückzukehren.

SAVE TO <Datei> ALL LIKE <Muster> oder ALL EXCEPT <Muster> – speichert alle temporären Variablen oder nur diejenigen, die durch <Muster> definiert sind in der angegebenen Datei. »?« wird benutzt, um einzelne (beliebige) Zeichen zuzulassen; mit " " sind beliebige Zeichenreihen erlaubt.
Beispiele:
(mit N1, N2, N10, V1, V2, V10 als Variablen):
. SAVE TO NUMBER ALL LIKE N (Speichert N1, N2, N10)
. SAVE TO NUMBER ALL LIKE N? (Speichert N1, N2)
. SAVE TO NUMBER ALL EXCEPT ?1 (Speichert N2, V2)

SELECT :PRIMARY / secondary: – schaltet zwischen primärer und sekundärer Datenbank um. Dadurch kann der Benutzer gleichzeitig mit zwei Dateien arbeiten, wobei der Satzzähler jeweils erhalten bleibt. Nach dem Start von dBase II ist zunächst die primäre Datenbank aktiviert.

SET – setzt dBase Kontrollparameter.
Syntax:
SET <Parameter> ON oder OFF
SET <Parameter> TO <Option>
Alle SET-Kommandos sind unten in alphabetischer Reihenfolge aufgelistet: Beachten Sie, daß einige SETs vom ON/OFF-Typ sind, andere aber bestimmte Eingaben erfordern. Standardwerte für ON/OFF-Kommandos werden jeweils durch Großbuchstaben (ON oder OFF) angezeigt.
Beispiel:
SET BELL :ON/off: - Standardwert ist ON.

SET ALTERNATE :OFF/on: – ON sendet alle Bildschirmausgaben (außer im Bildschirm-Modus) zu einer Diskettendatei (dem muß vorangehen: SET ALTERNATE TO <Datei>). OFF unterbricht die Ausgabe in die Datei.

SET BELL :ON/off: – Mit ON ertönt ein akustisches Signal, wenn ein ungültiges Zeichen eingegeben oder die Feldgrenze überschritten wird. OFF schaltet das Signal aus.

SET CARRY :OFF/on: – Mit ON werden Daten aus dem vorherigen Datensatz in den laufenden Satz geschrieben, wenn im Bildschirm-Modus APPEND benutzt wird. OFF läßt das Feld leer.

SET COLON :ON/off: – ON zeigt im Bildschirm-Modus Doppelpunkte an, um Variablen einzugrenzen. OFF hebt diese Anzeige auf.

SET CONFIRM :OFF/on: – ON verhindert im Bildschirm-Modus das automatische Springen zum näch-

sten Feld, wenn das laufende Feld gefüllt ist. OFF wartet auf die Eingabe von <RETURN>, bevor der Cursor zum nächsten Feld springt.

SET CONSOLE :ON/off: – Mit ON erscheinen alle Eingaben auf dem Bildschirm. OFF schaltet die Bildschirm-Anzeige aus; das System ist anscheinend »tot«.

> SET DEBUG :OFF/on: – Mit ON erscheint die mit ECHO und STEP erzeugte Ausgabe auf dem Drucker. OFF läßt die Ausgabe auf dem Bildschirm erscheinen.

SET ECHO :OFF/on: – ON ermöglicht die Verfolgung einer Programmausführung durch Anzeige aller Kommandos auf dem Bildschirm. OFF zeigt keinen Ausführungsbericht.

SET EJECT :ON/off: – ON veranlaßt den REPORT-Befehl, einen Seitenvorschub auszuführen, bevor ein auszugebender Bericht zum Drucker geschickt wird. OFF verhindert den Seitenvorschub.

SET ESCAPE :ON/off: – Mit ON kann der Benutzer die Ausführung der Programmdatei abbrechen, indem er die <ESC>-Taste drückt. OFF erlaubt keinen Abbruch mit der <ESC>-Taste.

SET EXACT :OFF/on: – ON erfordert genaue Eingaben der Zeichenketten in jeder Einzelheit (bei FOR<Ausdr>, FIND-Kommandos etc.). OFF erlaubt die Eingabe voneinander abweichender Zeichenketten, zum Beispiel unterschiedlicher Länge: 'ABCDEF' = 'ABC'.

SET INTENSITY :ON/off: – ON ermöglicht eine inverse Darstellung oder eine andere Helligkeit in der Bildschirmanzeige (falls die Hardware dies zuläßt). OFF schaltet diese Einrichtung ab.

SET LINKAGE :OFF/on: – ON erlaubt Bewegungen des Satzzeigers in der primären und sekundären Datenbank mit Hilfe von Kommandos, die einen bestimmten <Bereich> definieren, zum Beispiel Bewegungen nur abwärts. OFF schaltet diese Einrichtung ab.

SET PRINT :OFF/on: – ON sendet die Ausgabe zum Drucker. OFF stoppt die Druckerausgabe.

SET RAW :OFF/on: – Mit ON werden bei DISPLAY- und LIST-Befehlen keine Leerstellen zwischen den Feldern eingefügt; OFF fügt Leerstellen zwischen den Feldern ein.

SET SCREEN :ON/off: – ON erlaubt Bildschirmbearbeitung mit APPEND-, EDIT-, INSERT-, READ-, und CREATE-Kommandos.

SET STEP :OFF/on: – ON unterstützt das Testen von Programmdateien, indem diese in Einzelschritten ausgeführt werden, das heißt nach jedem Befehl angehalten wird. OFF schaltet diese Testeinrichtung ab.

SET TALK :ON/off: – ON zeigt Ergebnisse der Kommandoausführung auf dem Bildschirm an. OFF unterdrückt diese Anzeige.

SET ALTERNATE TO <Datei> – erzeugt eine Diskettendatei mit der Erweitung .TXT zur Protokollierung der Bildschirmausgaben. SET ALTERNATE TO schließt die .TXT-Datei.

SET COLOR TO <n1,n2> – auf den 8-Bit-Computern nicht verfügbar.

SET DATE TO <xx/xx/xx> – auf den 8-Bit-Computern nicht verfügbar.

SET DEFAULT TO <Laufwerk> – auf den 8-Bit-Computern nicht verfügbar.

SET FORMAT TO <SCREEN/ PRINT) – Mit SCREEN erfolgt die Ausgabe der SAY-Kommandos auf dem Bildschirm, mit PRINT auf dem Drucker.

SET FORMAT TO <Formatdatei> – öffnet eine .FMT-Datei, die dBase II benutzt, um den Bildschirm bei READ-, APPEND-, EDIT-, INSERT-, CREATE-, und SAY-Kommandos zu formatieren (Maske). SET FORMAT TO schließt die Formatdatei.

SET HEADING TO <Zeichenkette> – speichert <Zeichenkette> intern und druckt sie als Kopfzeile im Bericht.

SET MARGIN TO <n> – festlegen des linken Randes auf dem Drucker in Spalte <n>.

SKIP - <n> – bewegt den Satzzähler auf- oder abwärts um n Datensätze innerhalb der Datei. Standardwert ist +1.

SORT ON <Feld> TO <Datei> ASCENDING / DESCENDING – schreibt eine neue Kopie der Datei mit allen Datensätzen in einer bestimmten Reihenfolge. Benutzt den ASCII-Code für die Sortierung (Leerstellen, Zahlen, Großbuchstaben, Kleinbuchstaben und dann Symbole). Zum Löschen markierte Sätze werden übergangen. Standardwert für die Sortierfolge ist ASCENDING (aufsteigend).

Beispiel:
```
. USE ADRESSLISTE
. SORT ON PLZ TO POSTLEITZAHL
DESCENDING
```

STORE <Ausdr> TO <tempVar> – speichert den Wert eines Ausdrucks in eine temporäre Variable.

Beispiel:
```
. STORE 3 TO NUMMER
3
. STORE NUMMER + 9 TO NUMMER2
12
. STORE 'KLAUS' TO NAME
KLAUS
. ? NUMMER+NUMMER2, ' ', NAME
15 KLAUS
```

SUM – berechnet und zeigt die Summen von numerischen Feldern der Datei an. Die Option <Bereich> bestimmt die Menge der Datensätze für die Summierung; FOR <Ausdr> gibt für die Summierung bestimmte Bedingungen an; TO <tempVar> speichert das Ergebnis in der angegebenen temporären Variablen. Standardwert für <Bereich> ist ALL (alle). Zum Löschen markierte Datensätze werden übergangen.

Syntax:
```
SUM <Feld> :,<Feld2>: :<Bereich>:
:TO <tempVarliste>: :FOR <Ausdr>:
```

Beispiele:
```
. USE EINKAUFSLISTE :Felder sind
TEIL, ANZAHL, PREIS:
. SUM PREIS ANZAHL FOR TEIL =
'Nahrung'
. SUM ANZAHL FOR TEIL = 'Bier' TO
GETRÄNK
. SUM ANZAHL, ANZAHLPREIS FOR
TEIL = 'BIER' .AND. PREIS >1.00
```

TEXT – erzeugt programmierte Textausgaben. Der Text ist zwischen TEXT und ENDTEXT einzuschließen. Erspart die

(umständlichere) Verwendung von »?« beziehungsweise SAY.

TOTAL – erzeugt eine summarisch ausgewertete Form einer indizierten oder vorsortierten Datei. Der Schlüssel muß als Index vorhanden oder die Datenbankdatei danach sortiert sein. Es werden dann die angegebenen Felder der Sätze mit gleichem Schlüssel summiert und das Ergebnis in einer weiteren Datei abgelegt. Mit TOTAL kann man auch duplizierte Sätze entfernen.
Syntax:
TOTAL TO <Datei> ON <Schlüssel> :FIELDS <Feldliste>:

UPDATE – aktualisiert vorsortierte oder indizierte Dateien im Batch durch Übernahme von Informationen von (FROM) einer angegebenen Datei (mit demselben Schlüssel vorsortiert). Die Schlüssel der benutzten Datei werden mit den Schlüsseln der angegebenen Datei auf Übereinstimmung geprüft. dBase II kann dann zusammengehörige Felder aus beiden Dateien miteinander verbinden (zum Beispiel Addieren). Es können auch ganze Felder in der benutzten Datei durch Inhalte der entsprechenden Felder der anderen Datei ersetzt werden.
Syntax:
UPDATE FROM <Datei> ON <Schlüssel> :ADD <Feldliste>: :REPLACE <Feldliste> oder <Feld> WITH <Feldliste>: :RANDOM:

USE <Datei> INDEX <Indexdateiliste> – bestimmt die Datei für die nachfolgende Bearbeitung. USE schließt automatisch die mit einem vorherigen USE-Kommando geöffnete Datei. Optional kann ein Index vergeben werden, um die Datei zu ordnen. Weitere mit USE angeschlossene Indexdateien werden bei Änderungen automatisch aktualisiert.

WAIT TO <tempVar> – wird in Programmdateien benutzt, um die dBase II-Ausführung solange zu unterbrechen, bis auf der Tastatur ein einzelnes Zeichen eingegeben wird. WAIT TO <tempVar> speichert automatisch die Tastatureingabe in eine temporäre Variable und kann dadurch den Ablauf des Programms steuern.

Tabelle 6. Die dBase II-Fehlermeldungen

Meldung	Ursache/Beseitigung
UNZULÄSSIGE DEZIMALSTELLEN IM FELD	– Die Anzahl der Dezimalstellen bei der Feld-Definition ist neu einzugeben.
UNZULÄSSIGER DATEINAME	– Syntaxfehler im Dateinamen.
UNZULÄSSIGER FELDNAME	– Unzulässige Zeichen/Länge im Feldnamen; neu eingeben.
UNZULÄSSIGER FELDTYP	– Nur C (Character), N (numerisch) oder L (logisch) zugelassen.
UNZULÄSSIGE FELDLÄNGE	– Feldlänge minimal 1, maximal 255 Stellen. Bei Stringbefehlen darf die Stellenangabe den Wert 255 nicht übersteigen.
EINFÜGEN (INSERT) NICHT MÖGLICH – DATENBANKDATEI IST LEER	– INSERT durch APPEND ersetzen.
DATEI KANN NICHT GEÖFFNET WERDEN	– Prüfen, ob Datei vorhanden ist.
BEFEHLSDATEI NICHT VORHANDEN	– Dateiname/Laufwerksangabe prüfen.
DATENELEMENT NICHT GEFUNDEN	– Prüfen, ob Feldbezeichnung korrekt beziehungsweise das Feld in der Struktur vorhanden ist. Beim REPLACE-Befehl diesen neu eingeben.
DATENBANKDATEI WURDE OHNE INDEXDATEI AKTIVIERT	– FIND ist nur auf indizierten Datenbankdateien möglich. Die Index-Datei mit USE anschließen.
DISKETTENVERZEICHNIS IST VOLL	– Nicht benötigte Dateien löschen.
DISKETTE IST VOLL	– Nicht benötigte Dateien löschen.
DATEIENDE UNERWARTET ERREICHT	– Datenbankdatei weist Fehler auf, gegebenenfalls kopieren beziehungsweise reindizieren.
DIE »FIELD«-ANGABE FEHLT	– CHANGE-Kommando neu eingeben.
DATEI BEREITS VORHANDEN	– Nicht benötigte Datei gleichen Namens löschen oder anderen Namen verwenden.
DATEI NICHT VORHANDEN	– Prüfen, ob Datei vorhanden ist, etwa mit DISPLAY FILES LIKE *.*
DATEI IST BEREITS ERÖFFNET	– Mit USE oder CLEAR aktive Datei(en) abmelden.
FORMATDATEI KANN NICHT GEÖFFNET WERDEN	– ».FMT«-Datei prüfen, etwa durch Auflisten.
ES WURDE NOCH KEINE FORMATDATEI AUSGEWÄHLT	– Formatdatei aktivieren.
UNGÜLTIGER DATENTYP	– SORT kann nicht auf logischen Feldern sortieren.
UNGÜLTIGER WERT BEI »GOTO«	– Satznummer muß im belegten Bereich der Datenbank und im Bereich zwischen 1 und 65535 liegen.

UNGÜLTIGER VARIABLENNAME	- Nur alphanumerische Zeichen und Doppelpunkte sind als Namen für Felder und Variable zugelassen.
INDEXDATEI PASST NICHT ZUR DATENBANK	- Nur zur Datenbankdatei gehörende Indexdateien verwenden.
INDEXDATEI KANN NICHT GEÖFFNET WERDEN	- Dateiname prüfen oder Datenbank neu indizieren.
MIT JOIN WURDE VERSUCHT, MEHR ALS 65534 SÄTZE ZU ERZEUGEN	- Die »FOR«-Bedingung begrenzen.
SCHLÜSSEL HABEN ABWEICHENDE LÄNGE	- UPDATE verlangt übereinstimmende Schlüssel.
MAKRO IST KEINE ZEICHENFOLGE	- Makroexpansion verlangt Variablen mit Zeichenreihe als Wert.
MAXIMAL 5 FELDER FÜR SUMMENBILDUNG MÖGLICH	- SUM nur mit maximal 5 Feldern.
MAXIMALE SCHACHTELUNGSGRENZE ÜBERSCHRITTEN	- Programmdateien überarbeiten und zusammenfassen.
ES FEHLT DER NUMERISCHE AUSDRUCK ZUM SUMMIEREN	- SUM erfordert Angabe des/der zu summierenden Feldes(r).
DIE »FOR«-ANGABE FEHLT	- JOIN erneut mit »FOR«-Angabe eingeben.
DIE »FROM«-ANGABE FEHLT	- UPDATE erneut mit »FROM«-Angabe eingeben.
WERT NICHT GEFUNDEN	- Nicht unbedingt ein Fehler: Der Schlüsselwert ist nicht vorhanden (# = 0).
AUSDRUCK IST NICHT NUMERISCH	- SUM erfordert einen numerischen Ausdruck für die Summierung.
DATEI NICHT VORHANDEN	- Name prüfen, ggf. vorhandene Dateien auflisten lassen.
DIE »ON«-ANGABE FEHLT	- UPDATE oder INDEX mit »ON«-Angabe neu formulieren.
SPEICHER FÜR TEMPORÄRE VARIABLEN IST VOLL	- Nicht benötigte temporäre Variablen löschen.
SATZLÄNGE ÜBERSCHREITET MAXIMALWERT VON 1000 BYTE	- Feldlängen kürzen, so daß Satzlänge <= 1000 Byte.
SATZ NICHT IN INDEXDATEI ENTHALTEN	- Index-Datei nicht aktuell; Datenbankdatei reindizieren.
SATZ AUSSERHALB DES IN DATENBANKDATEI BELEGTEN BEREICHES	- Datei überprüfen (gegebenenfalls COPY) beziehungsweise reindizieren.
INTERNER FEHLER BEIM SORTIEREN	- Bitte Händler unterrichten
UNTERSCHIEDLICHE DATENTYPEN BEI QUELLE UND ZIEL	- Datentypen müssen übereinstimmen.
SYNTAXFEHLER	- unbekanntes Kommando, neu formulieren.
SYNTAXFEHLER BEI DER FORMAT-DEFINITION	- ein @SAY GET PICTURE wurde fehlerhaft formuliert.
DIE »TO«-ANGABE FEHLT	- Kommando mit »TO«-Angabe neu eingeben.
ZU VIELE ZEICHEN (BEFEHLSZEILE ZU LANG)	- Feldlänge oder maximale Länge der Befehlszeile wurden überschritten.
ZU VIELE DATEIEN ERÖFFNET	- Nur bis zu 16 Dateien (aller Typen) können gleichzeitig aktiviert werden.
ZU VIELE TEMPORÄRE VARIABLE	- Höchstens 64 temporäre Variable sind möglich; nicht benötigte Variablen löschen.
ZU VIELE »RETURNS« BENUTZT	- Struktur der Programmdateien überprüfen (insbesondere Schachtelung, Schleifen).
DIE »WITH«-ANGABE FEHLT	- REPLACE mit »WITH«-Angabe neu eingeben.
DATEI-NUMMER NICHT ZUGEORDNET	- Prüfen, ob DBASEMSG.TXT vorhanden ist
UNBEKANNTER BEFEHL	- Formulierung prüfen.
VARIABLE NICHT VORHANDEN	- Temporäre Variable erzeugen beziehungsweise Name korrigieren.

Tabelle 7. dBase II-Kenndaten und -Grenzwerte

Anzahl Felder/Datensatz	32
Anzahl Zeichen/Datensatz	1000
Anzahl Datensätze/Datei	65535
Anzahl Zeichen/Zeichenkette	254
Rechengenauigkeit	10 Stellen
größter Wert etwa	1.8×10^{63}
kleinster Wert etwa	1.0×10^{-63}
Anzahl temporäre Variable	64
Anzahl Zeichen/Kommandozeile	254
Anzahl <Ausdr> in SUM-Kommando	5
Anzahl Zeichen im REPORT-Kopf	254
Anzahl Felder im REPORT	24
Anzahl Zeichen im Indexschlüssel	99
Länge einer Programmdatei	unbegrenzt

Literatur:
Dr. Peter Albrecht, dBase II für den Commodore 128 PC, Markt&Technik Verlag 1985, ISBN 3-89090-189-1, 280 Seiten, Preis: 49 Mark
Dr. Peter Albrecht, Das Datenbanksystem dBase II, Markt&Technik Verlag 1985, ISBN 3-89090-143-3, 280 Seiten, Preis: 68 Mark
Wayne Ratliff, dBase II-Anwenderhandbuch, Ashton-Tate GmbH, Markt&Technik Software-Verlag 1984, 437 Seiten, Preis: 49 Mark

Die Floppy des Commodore 64 und VC-20

Schon nach kurzer Zeit merkt jeder, der sich mit dem VC-1541-Laufwerk beschäftigt, daß das mitgelieferte Handbuch nur äußerst dürftige Informationen über das Arbeiten mit der Floppy-Station und ihre Arbeitsweise gibt. So ist es erfreulich, daß rührige Verlage den vielen Floppy-Besitzern die Chance eröffnen, mehr über ihr Laufwerk zu erfahren und dieses optimal zu nutzen. Genau diesem Zweck dient dieses, von zwei Praktikern geschriebene Buch. Nach dem Vorwort versteht sich das Buch als Ergänzung zum Original-Handbuch. Gegebenenfalls wird deshalb zur Vermeidung von Wiederholungen auf das Handbuch verwiesen. Anleitungen für Wartungs- und Reparaturarbeiten werden nicht gegeben. Leicht verständlich, aber gründlich erläutern die Autoren dagegen die Arbeitsweise der VC 1541, die Basic-Befehle zur Diskettenverwaltung und zur Dateibearbeitung. Von besonderem Interesse dürften die Ausführungen über Direktzugriffe auf die Diskette und deren Anwendungen sein. Ausführlich gehen die beiden Autoren anschließend auf die Programmierung der VC 1541 in Maschinensprache und die damit verbundenen Vorteile ein. Dabei bleiben sie nicht in der Theorie stecken, sondern geben jeweils ein instruktives Beispiel für die praktische Anwendung der besprochenen Routine. Diese Programmbeispiele sind in 6502/6510-Assembler geschrieben. Aus der Vielzahl der vollständig abgedruckten Programme im Anhang seien hier besonders die Kopierprogramme für ganze Disketten und für einzelne Programmdateien, für sequentielle und relative Dateien sowie der Disk-Monitor erwähnt. Diese jeweils etwa 150 Zeilen langen Programme sind gut erklärt und in Basic geschrieben, enthalten aber zur Steigerung der Geschwindigkeit einige Maschinensprache-Routinen. Ärgerlich ist dabei nur, daß keine käufliche Diskette mit den vorgestellten Programmen vorliegt und dem Leser das mühsame Abtippen nicht erspart bleibt. Sonst ist das Buch aber jedem Floppy-Besitzer, der mehr als nur Programme laden und speichern möchte, zu empfehlen.

(Dieter Hein/bj)

Info: Dipl.-Phys. Dr. H. Riedel und Dipl.-Mathem. C. Hentschel, Die Floppy des Commodore 64 und VC-20, Friedrich Kiehl Verlag GmbH, 156 Seiten, ISBN 3-470-80431-1, Preis: 29,80 Mark

Die Floppy 1570/1571

Dieses im Markt&Technik Verlag erschienene Buch behandelt auf das ausführlichste die beiden Floppystationen 1570 und 1571 von Commodore. Das Besondere beider Laufwerke ist die Tatsache, daß sie mehrere Diskettenformate lesen und auch schreiben können, und gerade dieser Sachverhalt bildet den Schwerpunkt des Buches.

Neben den, auch in der Anleitung von Commodore stehenden, Tatsachen, die für den Einsteiger wichtig sind, widmet sich das Buch vor allem den Bereichen, die in sonst vorhandener Lektüre gar nicht oder nur unvollständig und schwer verständlich behandelt werden.

Das Kernstück des Buches ist dabei das ausführlich und gründlich dokumentierte DOS-Listing, das keine Wünsche mehr offenläßt. Der beschreibende Text ist so ausführlich, daß er auch ohne die nebenstehenden Assemblerbefehle zusammenhängend gelesen und verstanden werden kann.

Für den Anfänger beinhaltet das Buch alle wichtigen Grundlagen, die ausführlich und leicht verständlich erklärt sind. Der fortgeschrittene Programmierer findet in dem Buch alle wichtigen Einzelheiten, die ihn zum Profi werden lassen. Der Profi schließlich erhält ein rundum gelungenes Nachschlagewerk, das, mit vielen Tabellen und Verzeichnissen ausgestattet, ein hervorragender Helfer bei allen Problemen ist.

Insgesamt ein sehr sorgfältig zusammengestelltes Werk, das kaum noch Wünsche bezüglich Informationen zu den Laufwerken 1570 und 1571 offenläßt.

(Martina Müller/bj)

Info: Karsten Schramm, Die Floppy 1570/1571, Markt&Technik Verlag AG, Hans-Pinsel-Str. 2, 8013 Haar bei München, 470 Seiten, ISBN 3-89090-185-9, Preis: 52 Mark

Multiplan für den Commodore 128 PC

Hauptsächlich dem professionellen Anwender soll mit diesem Werk geholfen werden, einen schnellen und leichtverständlichen Einstieg in das Arbeiten mit Multiplan zu erhalten. Nach einer kurzen Einführung, in der die generellen Möglichkeiten des leistungsfähigen Kalkulationsprogramms beschrieben werden, wird der Anwender schrittweise in die Eigenheiten von Multiplan eingeführt. Nach einer allgemeinen Beschreibung der möglichen Felder wird im weiteren – aufbauend auf dem bisher Gelernten – weiter ins Detail gegangen. Nachdem auch die Handhabung von Formeln vermittelt ist, lernt der Anwender an Kalkulationsbeispielen sehr schnell, mit Multiplan umzugehen. Als vorteilhaft erweist sich der modulare Aufbau der Beispiele, das heißt, sobald neue Themen erschlossen werden, findet das vorhergehende Beispiel Verwendung und wird entsprechend erweitert. Für geübte Multiplan-Anwender steht am Ende des Buches eine detaillierte Funktionsübersicht zur Verfügung, die zusätzlich noch durch eine Kurzbeschreibung der CP/M-Befehle ergänzt ist. Für Anwender, die Multiplan konsequent einsetzen wollen, erweist sich das vorliegende Werk sicherlich bald als unentbehrliches Nachschlagewerk, egal ob Anfänger oder Profi.

(Martina Müller/bj)

Info: Dr. Peter Albrecht, Multiplan für den Commodore 128 PC, Markt&Technik Verlag AG, 226 Seiten, ISBN 3-89090-187-5, Preis: 49 Mark

dBase II für den Commodore 128 PC

In leicht verständlicher Form vermittelt der Autor anhand von Beispielen einen Einstieg in das Arbeiten mit dBase II. Die Vorstellung neuer Befehle erfolgt jeweils im Rahmen einer bestimmten Anwendung. Dadurch ist es möglich, sich weitaus schneller in die komplexen Gebilde »Dateiverwaltung und Dateiverwaltungssprache« einzuarbeiten. Von der Erstellung einer Datei, dem Anfügen und Ändern beliebiger Daten, über den gezielten Zugriff auf Daten und Datenfelder mit verknüpften Suchbedingungen, dem Ändern von Dateistrukturen, Sortieren, Indizieren und Erstellen von »Report«-Dateien bis hin zur Stapelverarbeitung mehrerer Dateien, Programmen in dBase II und der Kommunikation mit anderen CP/M-Programmen ist in diesem Buch alles enthalten,

was für den Einsteiger eine wertvolle Arbeitshilfe und den Fortgeschrittenen ein wichtiges Nachschlagewerk darstellt. Nach dem Durcharbeiten des Buches im direkten Dialog mit dem Programm ist man in der Lage, individuelle Anwendungen mit dBase II zu realisieren und mit dem vermittelten Wissen selbst die sehr komplexen Möglichkeiten zu nutzen, zumindest sich in diese Gebiete mit dem Handbuch von dBase II schnell einzufinden.

(Martina Müller/bj)

Info: Dr. Peter Albrecht, dBase II für den Commodore 128 PC, Markt&Technik Verlag AG 1985, 280 Seiten, ISBN 3-89090-189-1, Preis: 49 Mark

Die Floppy 1541

Zu den etablierten Standardwerken über »die Floppy 1541« gehört das gleichnamige Buch von Karsten Schramm, der den Lesern als Autor des Floppy-Kurses und als Redakteur des »64'er« kein Unbekannter ist. Hier plaudert ein Profi aus seiner Trickkiste. Denn bei diesem Floppy-Buch wurde der Schwerpunkt auf die Themen gelegt, bei denen andere aufhören.

Dennoch ist das vorliegende Buch nicht nur für Profis geeignet: Wer bisher nur die Befehle LOAD und SAVE mit seiner 1541 in Verbindung bringen konnte, der erfährt hier, wie man sequentielle, relative und Direktzugriffs-Dateien realisieren und verwenden kann.

Einige der weiteren Themen: Fehler im Commodore-DOS werden offengelegt, der serielle Bus wird unter die Lupe genommen und nach Hypra-Load-Manier beschleunigt. Methoden zur Rettung von verlorengegangenen Daten und fehlerhaften Blöcken werden vorgestellt. Und dies sind noch längst nicht alle der angesprochenen Bereiche. Das Allerbeste an diesem Buch ist allerdings das dokumentierte Listing des 1541-ROM. Praktisch jeder einzelne Maschinenbefehl wurde mit einem erläuternden Text versehen, weiter gibt es zu jeder der rund 400 Einzelroutinen des DOS einen kleinen einleitenden Text, dem dann die ausführliche Dokumentation neben dem Assembler-Listing folgt.

Die Dokumentation, die fast die Hälfte des Buches in Anspruch nimmt, wird von einer ebenso ausführlichen RAM-Belegung ergänzt. Mehrere nützliche Programme und ein ausführliches Stichwortverzeichnis runden das äußerst positive Gesamtbild ab. Die Diskette mit allen abgedruckten Programmen kann zu einem Preis von 29,80 Mark separat bestellt werden. Das eindeutige Urteil: Ein Floppy-Buch, das im Bücherschrank eines 1541-Besitzers nicht fehlen sollte.

(Boris Schneider/ev)

Info: Karsten Schramm, Die Floppy 1541, Markt&Technik 1985, 434 Seiten, ISBN 3-89090-098-4, 49 Mark, Preis der Diskette zum Buch: 29,80 Mark.

C 1571 & 1570: Das große Floppybuch

Das im Buchtitel enthaltene Prädikat »groß« verdient dieses Buch von Data Becker sicherlich, denn mit 583 Seiten ist es ein umfangreiches Werk.

Die große Stärke des Buches ist das rund 350 Seiten umfassende DOS-Listing, das zudem eine Cross-Reference enthält. Es ist sehr gewissenhaft dokumentiert und glänzt durch gute optische Aufmachung. Anwendungen werden direkt anhand von Beispielen erklärt. Unter anderem befindet sich im Buch ein Programm zum Analysieren von MFM-Formaten, welches Informationen über die Anzahl der Byte pro Sektor, der Tracknummern und ähnlichem liefert. Da dieses und ähnliche Programme in Basic geschrieben sind, lassen sie sich auch für »Nicht-Assembler-Programmierer« nachvollziehen und gegebenenfalls ändern. Die Neuauflage dieses Buches wurde hinsichtlich der Rechtschreibung überarbeitet. Die diesbezüglich geäußerte Kritik in Ausgabe 4/86 ist also gegenstandslos geworden. Weiterhin sind zur gleichen Ausgabe zwei Berichtigungen vorzunehmen: Der Preis des Buches wurde fälschlicherweise mit 69 Mark angegeben. Der tatsächliche Preis beläuft sich jedoch auf 49 Mark, der Preis der Diskette mit Beispielprogrammen 29 Mark. Abschließend ergibt sich ein positiver Gesamteindruck, sowohl hinsichtlich des Preis-Leistungsverhältnisses, als auch der Fülle von Informationen.

(Florian Müller/ev/bj)

Info: Rainer Ellinger, C1571&1570: Das große Floppybuch, Data Becker, 583 Seiten, ISBN 3-89011-124-6, Preis: 49 Mark, Diskette mit Beispielprogramm: 29 Mark

Die Dateiverwaltung für den C 64 & C 128

»Die Dateiverwaltung für den C64 & C128« enthält eine gelungene Mischung aus Lehrbuch und Software. Das durch Menütechnik benutzerfreundliche Dateiverwaltungsprogramm ist übersichtlich in Unterprogrammen geschrieben. Das Buch erläutert ausführlich und gut verständlich das Prinzip der Datenverwaltung und alle Teile der in Basic und Assembler geschriebenen Routinen. Durch Verwendung einer Index-sequentiellen Datei wird schnelles Durchsuchen möglich. Vorgestellt wird eine Adreßverwaltung, die vorgeschlagene Eingabemaske kann aber leicht abgeändert werden. Dem Leser wird empfohlen, vorgegebene Routinen in eigene Programme zu übernehmen. Nette Ratschläge, was dabei besonders zu beachten ist, machen solche Vorhaben leicht. Erfreulich ist es, daß für das Programm auf dem C 128 die Vorteile des komfortableren 7.0-Basic verwendet wurden. Besonderen Aufwand erforderte das Neuschreiben der Assemblerroutinen wegen des Bank-Switchings. Leider ist keine Möglichkeit vorgesehen, auf dem Bildschirm oder beim Druck die deutschen Umlaute und »ß« auszugeben. Gerade für ein Adressenverwaltungsprogramm ist diese Option jedoch unabdingbar. Vermißt wird auch ein Hinweis, ob und wie die erstellten Dateien von anderen Programmen (Textverarbeitung?) benutzt werden können.

Die zum Buch erhältliche Diskette kann zu einem Preis von 29 Mark einzeln bestellt werden.

Mit dem Programm läßt sich recht gut und schnell arbeiten. Was einen stört, kann man wegen der genauen Erklärungen selbst verbessern. Für die eigene Programmierung kann der Leser viel lernen und schöne Routinen übernehmen.

(D. Hein/ev/bj)

Info: Said Baloui: »Die Dateiverwaltung für den C 64 & C 128«, Data-Becker, 272 Seiten, ISBN 3-89011-103-3, Preis 39 Mark, Diskette 29 Mark

Floppy VC 1541

Schon mancher Besitzer der Commodore-Floppy-Station hat mit der 1541 Ärger gehabt. Zu anfällig ist die Mechanik. Reparaturen in der Werkstatt dauern lange und kosten viel Geld. So ist es kein Wunder, daß viele 1541-Besitzer versuchen, notwendige Pflege- und Reparaturarbeiten selbst durchzuführen. Daß Eingriffe in die Floppy-Station nur nach Ablauf der Garantiezeit und von Leuten, die Erfahrung im Umgang mit hochempfindlichen Geräten haben, gemacht werden dürfen, kann nicht oft genug betont werden. Die vermeintliche Einsparung könnte sich leicht ins Gegenteil verkehren.

Die in den Kapiteln 1 und 2 beschriebenen Pflege- und Überwachungsarbeiten können als Wartung mit Hilfe der genauen Anleitungen und der vielen Zeichnungen im Buch von jedem, der nicht zwei linke Hände hat, ausgeführt werden. Das dritte Kapitel ist dem Justieren des Schreib-/Lesekopfes gewidmet. Sehr genau wird dabei geschildert, wie mit Hilfe des auf der Demo-Diskette befindlichen Programmes »Display T&S« geprüft werden kann, ob der Stopanschlag für die Spur 1 richtig eingestellt ist und ob sich der Schreib-/Lesekopf über alle Spuren korrekt bewegt. Mit Hilfe des Buches muß man den Stopanschlag gegebenenfalls um 0,1 bis 0,35 mm verstellen, noch schwieriger ist die Justage des Schreib-/Lesekopfes, obwohl das Buch genau beschreibt, wie man dabei mit oder ohne Oszilloskop vorgehen muß. Vier Programmlistings für Programme zum Erstellen einer Spureinstellungsdiskette, zur Spureinstellung, für einen Schreib-/Lesetest und zur Geschwindigkeitseinstellung können abgetippt werden und helfen dann bei den angeführten Arbeiten.

Für den Fachmann sehr nützlich ist die ausführliche Darstellung der Elektronik und Digitaltechnik.

(D. Hein/ev)

Info: Reinhold Herrmann, Floppy VC 1541, Data Becker 1985, 220 Seiten, ISBN 3-89011-079-7, Preis 49 Mark

Fragen und Antworten

Autostart

Wie bringe ich meine Programme dazu, daß sie nach dem Laden von Diskette oder Kassette automatisch starten? Carsten Bruch

Um einen Autostart bei Programmen auf Diskette hervorzurufen, genügt es, folgende Zeile einzugeben:

```
LOAD"NAME",8:
<SHIFT+RUN/STOP>.
```

Dabei müssen die Tasten <SHIFT+RUN/STOP> gleichzeitig gedrückt werden. Das Programm »NAME« wird dann ohne Eingabe von <RETURN> automatisch geladen und gestartet. Arndt Grothoff

Bits hörbar machen

Ist es beim C64 möglich, beim Laden von Datasette die Signale hörbar zu machen? Werner Frings

Um die Signale, die von der Datasette kommen, hörbar zu machen, muß einfach mit POKE 54296,15 die Lautstärke im SID auf 15 gesetzt werden. Nun sind die Signale beim Laden, Saven und Verifyen über den Lautsprecher des Monitors zu hören (auch bei Fast- und Turbo-Tape). Stefan Gossens

Disketten beidseitig verwenden?

Wenn man mit einem Bürolocher auf einer einseitig beschreibbaren Diskette am linken Rand eine Stanzung in Höhe des Schreibschutzes am rechten Rand vornimmt, läßt sich die Diskette beidseitig auf einer Floppy 1541 verwenden. Ist die Datensicherheit dabei gewährleistet, oder spricht etwas gegen diese Methode? Dipl.-Ing. M. Lohse

Bei einseitig beschreibbaren Disketten ist auch nur eine Seite durch den Hersteller geprüft. Die zweite Seite kann daher unter Umständen fehlerhafte Sektoren enthalten, die aber vom Floppy-Betriebssystem dann nicht benutzt werden. Wenn Sie nach dem Formatieren beim Listen der Directory die Meldung »664 Blocks free« sehen, sollte jedoch die Datensicherheit einigermaßen gewährleistet sein.

Für sehr wichtige Aufzeichnungen empfiehlt sich jedoch immer die Verwendung doppelseitig geprüfter Disketten.

Steckmodule speichern?

Wer kann mir mitteilen, wo der Modulbereich meines Commodore 64 liegt und ob es eine ähnliche Routine zum Abspeichern dieses Bereichs auf Diskette gibt? Hartmut Götze

Beim VC 20 liegt der Modulbereich von $A000 bis $BFFF und läßt sich mit folgender Eingabe auf Diskette kopieren:

```
POKE 43,0 : POKE 44,160 :
POKE 45,0 : POKE 46,192 :
SAVE " Name",8
```

Der Steckmodulbereich beim Commodore 64 liegt von $8000 bis $9FFF. In der obengenannten SAVE-Routine muß daher der POKE-Wert 160 durch 128 und der Wert 192 durch 160 ersetzt werden. Allerdings haben die Hersteller von Steckmodulen in der Regel einige Sicherungen gegen unerlaubtes Kopieren eingebaut.

1541 am C128

Kann man die alte 1541-Floppy und einen C64-Drukker (mit Interface) am C128 weiterverwenden? Jürgen Ruppert

Auch der serielle Bus des C128 ist aufwärtskompatibel konzipiert. Das bedeutet, alle Peripheriegeräte, die am seriellen Bus des C64 funktionieren, arbeiten auch am C128 einwandfrei. Allerdings wird der serielle Bus dann in einer langsamen Betriebsart (der gleichen wie beim C64) betrieben, und man muß auf die Vorteile des schnelleren Datentransfers verzichten. Peripheriegeräte, die speziell für den C128 entwickelt wurden (wie zum Beispiel die 1571-Floppy), können die schnelle Betriebsart des seriellen Busses nutzen. Die Ladezeit für eine hochauflösende Grafikseite (33 Blöcke auf Diskette) beträgt beispielsweise beim alten 1541-Laufwerk stattliche 25 Sekunden, bei der neuen 1571-Station dagegen nur noch runde 3 Sekunden.

Das Betriebssystem CP/M läuft sowieso nur in Verbindung mit der neuen Floppystation.

Fazit: Sie können die gute alte 1541-Floppy und den Drucker vom C64 übernehmen, verlieren dadurch aber insbesondere hinsichtlich der Floppystation so viele der Vorteile des C128-Systems, daß Sie sich ernsthaft fragen sollten, ob Sie überhaupt noch einen C128 brauchen.

Wordpro 3+ mit 1541?

Ich besitze einen C64 mit Floppy-Laufwerk 1541. Als Textverarbeitungsprogramm wollte ich Wordpro 3+ einsetzen. Dieses Programm scheint aber nur mit Druckern zu arbeiten, die über den IEC-Bus angesteuert werden. Wer weiß, wie ein eigener Druckertreiber in Wordpro eingebunden werden kann (Centronics-Schnittstelle über User-Port)? Eine andere Möglichkeit wäre die Umwandlung IEC-seriell/parallel. Wer hat so etwas schon realisiert? Heinz-Josef Erken

Es gibt 2 Versionen des Wordpro 3+ für den C64. Eine für den VC 1526, die zweite für andere (Commodore-)Drucker.

Datasette oder Diskette?

In welchem Umfang kann – ich bin Anfänger – die Datasette ein Diskettenlaufwerk ersetzen? Hans Georg Walther

Beide sind Massenspeicher – wem das Diskettenlaufwerk zu teuer ist, kann (und muß) ein Kassettenlaufwerk, also die Datasette, nehmen. Das Kassettenlaufwerk ist deutlich langsamer als das Diskettenlaufwerk. Dazu kommt ein zweiter Nachteil: Auf dem Magnetband in der Kassette werden die Daten sequentiell, also der Reihe nach hintereinander, gespeichert. Es kommt also beispielsweise die Adresse von Herbert Adam, dann die von Erich Berger und so weiter. Wenn Sie die Adressen in der Reihenfolge brauchen, in der sie gespeichert sind, funktioniert das ganz gut. Wenn Sie – bei alphabetischer Reihenfolge – erst die Adresse von Weber, dann die von Berger, dann die von Müller und so weiter brauchen, muß das Band immer erst zu der entsprechenden Stelle vor- beziehungsweise zurücklaufen, was äußerst zeitaufwendig ist. Außerdem macht es einige Arbeit, in eine gegebene Reihenfolge später neue Adressen einzufügen. Würde als Speichermedium eine Diskette verwendet, so könnte auf jede Adresse direkt zugegriffen werden (sogenannter Random Access), ohne daß ein Vor- oder Rücklauf des Bandes abgewartet werden muß.

Außerdem müssen Sie auf der Diskette die Daten nicht unbedingt in einer ganz bestimmten Reihenfolge speichern (Details sind von der Dateiorganisation und damit der Software abhängig). Faustregel: Mit einem Diskettenlaufwerk arbeitet man schneller und bequemer als mit einem Kassettenlaufwerk. Alle Anwendungen, die sich mit einem Kassettenlaufwerk realisieren lassen, sind auch mit einem Diskettenlaufwerk zu verwirklichen, während es umgekehrt eine Reihe von Anwendungen gibt, die nur mit einem Diskettenlaufwerk sinnvoll zu bewältigen sind.

Ärger mit der 1541

Ich habe Schwierigkeiten mit meinem Floppy-Laufwerk. Wenn es längere Zeit in Betrieb und ziemlich warm geworden ist, nimmt die Diskette kein Programm mehr an. Beim Speichern meldet die Floppystation dann einen Fehler 20 oder 27. Das Laufwerk lädt dann auch nicht mehr, auch keine Directories von anderen Disketten. Nach längerem Abkühlen kann wieder geladen werden, aber nicht mehr die Programme, die ich ich vorher abspeichern wollte. Auf dem Laufwerk ist noch Garantie und es war auch schon zur Reparatur beim Händler. Aber der Fehler tritt immer wieder auf. Schadet es der Floppysta-

tion, wenn ich einen Reset auslöse? Dabei läuft das Laufwerk kurz an und die rote Lampe leuchtet. Josef Spiertz

Ihr Problem beruht auf der thermischen Ausdehnung von Metallen. Die Mechanik des Laufwerks dehnt sich normalerweise aus. Bei Ihnen scheint der unglückliche Extremfall vorzuliegen, daß durch diese Ausdehnung der Schreib/-Lesekopf total verstellt wird. Dann kann die Floppystation weder Daten lesen noch schreiben. So, wie bei Ihnen geschildert, darf dieses Problem aber unter normalen Umständen nicht auftreten. Normalerweise ist ein mehrstündiger Betrieb, solange nicht laufend formatiert wird, ohne Probleme möglich. Es gibt hier zwei Lösungsmöglichkeiten: Sie sorgen für Kühlung, indem Sie das Laufwerk »offen«, das heißt ohne Gehäusedeckel, betreiben oder einen Lüfter verwenden. Die Alternative wäre, daß Sie auf einen Umtausch des Laufwerks bei Ihrem Händler bestehen.

Bei einem Reset am Computer wird auch in der Floppystation ein Reset ausgelöst. Das hat keinerlei schädliche Folgen für Floppystation und Diskette.

Unsichtbar nachladen?

Wie schaffe ich es, in Basic 2.0 ein Nachlade-Programm zu konzipieren, das bei entsprechender Menüwahl ein Teilprogramm nachlädt – und zwar ohne Ausgabe der Meldungen »Searching for« und »Loading«? Ipek Cüneyt

Beim Nachladen von Programmen müssen Sie folgendes speziell beachten: Der Befehl »LOAD« in einem Programm bewirkt nicht nur, wie im Direktmodus, das Laden eines Programms, sondern gleichzeitig einen Autostart. Falls das nachgeladene Programm größer ist als das ursprüngliche Programm, dann werden alle Variablen gelöscht. Sie sollten daher dafür sorgen, daß Ihr Hauptprogramm immer größer ist als das nachzuladende. Auch sollten Sie beachten, daß Strings in der Regel nicht mit übernommen werden. Wollen Sie Stringvariable dennoch ins nächste Programm übernehmen, dann müssen Sie beispielsweise im ersten Programm schreiben:

```
A$="HALLO"+""
```

Dies ist deshalb nötig, weil nur Stringvariable, die mit irgendeiner Stringoperation (+, LEFT$, MID$, RIGHT$) verknüpft wurden, ihren Wert auch im nachgeladenen Programm behalten. Um die manchmal unerwünschte Systemmeldung wie »Searching for« am Bildschirm nicht sichtbar werden zu lassen, gibt es eine sehr einfache Methode: Setzen Sie vor den LOAD-Befehl im Programm einfach die Schreibfarbe auf die Hintergrundfarbe. Sie verhindern damit zwar nicht die Ausgabe dieser Meldung, sie erscheint aber nicht sichtbar auf dem Bildschirm.

Einschaltprüfung für Floppy und Drucker?

Welche POKE-Adressen können in einem Basic-Programm zur Abfrage dienen, ob der Drucker oder die Floppystation eingeschaltet ist? Damit könnte ein Programmabbruch durch »Device not present« verhindert werden. Oskar Greifenberger jun.

Die etwas umfangreichere Lösung dieses Problems finden Sie auf Seite 77 in der Ausgabe 8/84 des 64'er Magazins.

Nochmal: Disketten doppelt beschreiben?

Bei uns geht das Gerücht um, daß es bei der Floppy 1541 eine Andruckrolle oder ähnliches gibt, die direkt auf die Oberfläche der B-Seite einer Diskette drückt. Die Anwendung von doppelbeschreibbaren Disketten ist somit schädlich für den Schreib-Lesekopf, da Schmutz übertragen wird. Stimmt das? Arndt Böhme

Leider ja. Der Schreib-Lesekopf der Diskette, der die Unterseite der Diskette »abtastet«, kann die Diskette nicht ohne mechanische Unterstützung lesen. Ein Andruckfilz drückt die Diskette deswegen von oben auf den Schreib-/Lesekopf. Wenn der Filz arg verschmutzt ist, kann es zu physikalischer Beschädigung sowohl von Diskette als auch vom Laufwerk kommen. Disketten, die auch nur minimal verschmutzt sind, sollten ersetzt werden. Bewährt haben sich auch Sicherheitskopien für den Fall des Falles. Ein Öffnen des Gerätes, um den Zustand des Kopfes zu überprüfen, beziehungsweise das gelegentliche Reinigen mit einer Reinigungsdiskette wäre wohl sinnvoll.

Floppy-DOS ändern?

Ist es möglich, sich (mit einem entsprechenden Monitorprogramm) das eingebaute DOS 2.6 der Floppy 1541 anzusehen und auch abzuändern? B. Heiler, J. Hamberger

Die Antwort lautet: Im Prinzip ja. Das DOS des 1541-Laufwerks läßt sich relativ einfach über den »M-R«-Befehl und das allgemein bekannte POKE in den Speicher holen. Dann geht die Analyse mit jedem Monitor, zum Beispiel unserem SMON. Ändern kann man es freilich nur auf die harte Methode, das heißt mit einem EPROM-Brenner. Kleine Veränderungen oder Zusatzroutinen können aber auch im RAM des Floppy-Laufwerks abgelegt werden. Hier verweise ich Sie auf unseren Floppykurs. Ein analysiertes DOS-Listing finden Sie im Buch »Die Floppy 1541« von Markt & Technik.

Zwei Datasetten am Commodore 64?

Auf der internen Platine der Datasette befindet sich eine Steckerleiste, auf die der Datasettenstecker haargenau paßt. Läßt sich darüber eine zweite Datasette anschließen? Uwe Bilo

Die Pin-Belegung des Platinensteckers in der 1530/C2N ist folgende:

1 +5V
2 Read Data
3 Write Data
4 Masse (GND)
5 Head hot
6 Head return

Während an den Pins 2 und 3 die gleichen Signale wie am computerseitigen Stecker liegen, sind die Pins 5 und 6 direkt mit dem Tonkopf verbunden. Diese Anschlüsse sollte man also mit Vorsicht behandeln. Sie sind vor allem interessant für Elektroniker, die ein Oszilloskop zur Hand haben und damit einen verstellten Tonkopf wieder justieren können.

Es ist möglich, eine zweite Datasette anzuschließen, wenn man einfach die entsprechenden Anschlüsse der Datasetten parallel auf den Kassetten-Port legt. Da jedoch weder die Datasette von sich aus über eine bestimmte Geräteadresse verfügt, noch der C64 zwei verschiedene Geräteadressen für Datasetten bereitstellt, wird man dabei allerdings auf folgende Probleme stoßen:

1. Beim Laden/Speichern mit einer Datasette läuft automatisch auch der Motor der anderen mit.
2. Der Computer kann nicht unterscheiden, an welcher Datasette eine Taste gedrückt ist. Die Entscheidung, welche Datasette nun angesprochen werden soll, liegt also immer beim Benutzer, der Computer spricht stets beide gleichzeitig an.
3. Der C64 hat natürlich trotz allem nur einen Kassettenpuffer, es ist also beispielsweise nicht möglich, auf der einen Datasette eine Datei zum Lesen zu öffnen und auf der anderen eine Datei zu schreiben.

Um Komplikationen zu vermeiden, wäre es nötig, die Motorsteuerung über den User-Port vorzunehmen und auch das Betriebssystem per Maschinenprogramm an die geänderten Verhältnisse anzupassen. Der Anschluß einer zweiten Datasette ist also nicht so einfach, wie es auf den ersten Blick aussieht. Bertram Dunskus

Laden mit Bild?

Während der C64 ein Programm von Kassette lädt, zeigt er normalerweise kein Bild auf dem Monitor an. Nun habe ich aber an einigen Maschinensprache-Programmen gesehen, daß es doch möglich ist, ein Bild während des Ladens stehen zu lassen. Ist dieser Effekt auch in Basic zu erzielen oder ist dazu die Kenntnis von Maschinensprache notwendig? Dieter Kurbjuhn

Vom Basic aus ist das nicht möglich. Da der Videochip und der Prozessor abwechselnd auf denselben Datenbus zugreifen, kann es bei Ladevorgängen von Kassette zu Zeitproblemen kommen, die das System zum Absturz bringen könnten. Um das zu verhindern, schaltet das Betriebssystem während der Dauer des Kassettenzugriffs den Videochip einfach aus,

wodurch natürlich das Bild verschwindet. Um das zu verhindern, darf man nicht auf die vorhandenen Betriebssystemroutinen zugreifen - wie es das Basic automatisch macht, sondern muß sich eigene Laderoutinen schreiben. Das allerdings ist nur in Maschinensprache möglich.

Diskette versehentlich formatiert?

Ich habe versehentlich eine Diskette mit vielen Programmen darauf formatiert. Wie kann ich diese Programme wieder zurückholen?

Hinnerk Behn

Wenn Sie die Diskette ohne Angabe einer ID formatiert haben (die Formatierung kann dann nur einige Sekundenbruchteile gedauert haben), dann ist das Directory gelöscht. Mit einem Diskettenmonitor (und dem Floppy-Kurs) könnten Sie bei etwas Erfahrung in der Lage sein, das Inhaltsverzeichnis wieder zu restaurieren. Da in der Regel aber mit ID formatiert wird, sind Ihre Programme wohl verloren. Das 1541-Laufwerk beschreibt nämlich beim normalen Formatieren alle Sektoren der Diskette mit Null-Bytes. Die ursprünglich darin enthaltene Information geht dabei genauso verloren wie beim Überspielen einer normalen Musik-Kassette.

»Load Error« bei Datasette?

Ich habe folgendes Problem mit meiner Datasette: Sie macht andauernd »? Load Error«, trotz Kassettenwechsel. Mit meiner Datasette ist alles in Ordnung. Man hat mir nur gesagt, daß vielleicht etwas mit meinem Mikroprozessor im C64 nicht stimmt. Kann es tatsächlich daran liegen?

Ronny Gaab

Leider kann man bei so allgemeinen Angaben nur vage Vermutungen über die Fehlerursache anstellen. Tritt der Lesefehler nur bei fremden Programmen auf oder auch bei selbst gespeicherten? Wurde die Speicherung mit VERIFY überprüft, und mit welchem Ergebnis? Können nur vereinzelt Programme nicht gelesen werden oder funktioniert gar nichts mehr?

Jedenfalls ist es sehr unwahrscheinlich, daß der Fehler am C64 liegt (an der CPU kann es schon gar nicht liegen, wenn der Computer sonst einwandfrei funktioniert).

Die häufigste Fehlerursache ist ein verschmutzter oder verstellter Tonkopf an der Datasette. Reinigen Sie Tonkopf, Bandführung und Andruckrolle von Zeit zu Zeit mit einem in Spiritus getränkten Wattestäbchen. Die korrekte Einstellung des Tonkopfes kann Ihr Fachhändler vornehmen; wenn Sie sich selbst daranwagen wollen, sollten Sie entsprechende Fachliteratur zu Rate ziehen.

Bei der Aufzeichnung denken Sie bitte unbedingt immer an das VERIFY, das nach jedem Abspeichern auf Kassette durchgeführt werden sollte. Verwenden Sie keine Billig-Kassetten, aber auch keine Chromdioxid- oder Reineisen (Metall)-Bänder. C 90- und C 120-Kassetten sollten Sie ebenfalls vermeiden. Lassen Sie am Anfang und Ende jeder Kassette mindestens 10 bis 20 Sekunden Band frei, da diese Bandstellen durch das ständige Anschlagen beim Umspulen besonders starken mechanischen Belastungen ausgesetzt sind. Achten Sie darauf, daß das Band in der Kassette frei beweglich ist (glatte Spulenwickel), eventuell ein paarmal vollständig vor- oder zurückspulen. Beachten Sie schließlich den obersten Programmierer-Grundsatz: Von jedem wichtigen Programm eine Sicherheitskopie anfertigen.

Ein Bug in der 1541?

Wenn ich ein Programm habe, das mit sequentiellen Files arbeitet, und ein solches Programm nach dem Speichern eines Files editiere, und es dann mit SAVE "@:Name",8 speichere, habe ich statt des Programms die Daten des sequentiellen Files auf der Diskette stehen. Woher kommt dieser sehr fatale und ärgerliche Fehler und wie kann man ihm abhelfen?

Alexander Picke

Es handelt sich hierbei um einen Fehler im DOS des 1541-Laufwerkes. Durch ein Speichern mit »@:Name« kann es aufgrund dieses Fehlers zu einer Veränderung des Blockverbindungsanzeigers auf der Diskette kommen - es wird auf ein anderes File oder einen leeren Sektor der Diskette zugegriffen. Dem Fehler abhelfen können Sie, indem Sie folgendermaßen vorgehen: Speichern Sie Ihr Programm nicht auf die »kaputte« Art, sondern löschen Sie erst die vorhergegangene Version von der Diskette. Speichern Sie erst dann Ihre neueste Programmversion (ohne den Klammeraffen) ab.

1-Mega-Byte-Floppy für C64?

Ich benötige für meinen C64 einen Massenspeicher mit wesentlich höherer Kapazität, als sie die 1541 bietet.

Joachim Kaluza

Eine Möglichkeit wäre die SFD 1001 von Commodore. Sie hat eine Speicherkapazität von rund 1000 KByte. Zusätzlich ist eine IEEE 488-Schnittstelle erforderlich (siehe 64'er Sonderheft 10/86)

DOS 5.1

Leider kann man bei den meisten kommerziellen Programmen nichts mit dem DOS 5.1 anfangen, da man aus diesen Programmen nur durch Abschalten des Computers wieder herauskommt. Aber dann ist auch das DOS 5.1 verloren, und es jedesmal neu zuladen ist doch zu umständlich. Kann man da nichts dran machen?

Heinrich Carstensen

Das DOS 5.1 ist eigentlich nur bei der Programmentwicklung nützlich. Es erleichtert das Arbeiten mit der Floppystation 1541, indem es Abkürzungen verwendet. Wenn man Spiele oder andere kommerzielle Software benutzt, ist die Anwendung des DOS sowieso wenig sinnvoll.

Daten speichern, ohne Datei zu laden?

Wie kann ich Daten auf Diskette speichern, ohne die Datei zu laden, umzuändern und neu zu speichern (das ist mir zu umständlich)?

Kurt Müller

Mit einer relativen Datei ist das ohne Schwierigkeiten möglich. Im Sonderheft 2/86 des 64'er-Magazins wurde sie ausführlich beschrieben. Und wem diese Methode noch zu umständlich erscheint, weil ja noch das Programm geladen werden muß, kann sich dieses Programm ja auf EPROM brennen. Dann genügt ein Einstecken des Moduls in den Expansion-Port, Computer einschalten und schon können die ersten Daten am Bildschirm bearbeitet werden.

Merge-Problem

Wie kann ich zwei Programme im Speicher zu einem einzigen zusammenfügen?

Martin Hossdorf

Auch ohne Basic-Erweiterung kann man beim C64/VC 20 Programme relativ einfach »mergen«. Erstes Programm laden und danach »POKE 43, PEEK(45) -2: POKE 44, PEEK(46)« eingeben. Dies schützt den ersten Programmteil. Jetzt kann man ein zweites Programm laden, nochmals die gleichen POKE-Befehle geben, ein drittes Programm laden und so fort. Nach »POKE 43,1 : POKE 44,8« stehen alle Programme zu einem einzigen zusammengefaßt im Speicher. Man sollte allerdings darauf achten, daß die nachgeladenen Programme höhere Zeilennummern haben, da es sonst zu Problemen kommen kann. Michael Aßfahl

Ersatz für 1541?

Wie kann ich ein BASF-Floppy-Laufwerk 6108 an den C64 anschließen?

Peter Bosse

Gibt es eine Möglichkeit, handelsübliche Floppy-Laufwerke beispielsweise von Teac, Sony oder BASF an den C64 anzuschließen? Wo könnte man Baupläne sowie Unterlagen über das Betriebssystem bekommen? Heinz Sigrist

Commodore-Laufwerke verfügen über eigene »Intelligenz«, handelsübliche Laufwerke sind dagegen »dumm«, so daß für ihren Einsatz, abgesehen von einem Betriebssystem, noch ein an den C64 angepaßter Controller nötig wäre. Ein Selbstbau dürfte sich kaum lohnen; unabhängige Anbieter Commodore-kompatibler Laufwerke beziehungsweise Controller sind uns nicht bekannt.

Checksummer 64 V3

Der Checksummer 64 V3 überprüft jede Basic-Zeile direkt nach der Eingabe, erkennt Fehleingaben sowie Vertauschungen von Ziffern und erspart eine aufwendige Fehlersuche.

Der Checksummer 64 V3 (Listing) ist ein kleines Maschinenprogramm, das Sie sofort unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.

So gehen Sie vor:

1. Programm abtippen und speichern.

2. Starten mit RUN.

3. Nach kurzer Zeit sehen Sie am Bildschirm: »CHECKSUMMER 64, CHECKSUMMER AKTIVIERT, AUSSCHALTEN MIT POKE 1,55, ANSCHALTEN MIT POKE 1,53, READY«.

4. Einschalten des Checksummers 64 V3 mit POKE 1,53.

5. Test: Geben Sie in einer freien Zeile ein: »1 REM« und drücken die <RETURN>-Taste. Am Bildschirm oben links sollten Sie die Prüfsumme <63> sehen.

6. Geben Sie ein Listing aus unserem Heft ein. Nach jeder Zeile wird die Zahl, die im Listing in Klammern < > steht, in den Bildschirm eingeblendet. Stimmen die Zahlen nicht überein, so liegt vermutlich ein Eingabefehler vor.

Die Zahl in den Klammern, und auch die Klammern selbst, dürfen beim Abtippen nicht eingegeben werden!

7. Der Checksummer 64 V3 bemerkt auch Vertauschungen von Zahlen und Buchstaben, aber nicht das Fehlen (oder Hinzufügen) von Leerzeichen.

8. Unsere Basic-Listings enthalten keine Steuerzeichen mehr. Diese werden ersetzt durch Klartext und stehen zwischen geschweiften Klammern. Deshalb sind weder die Klammern noch das was dazwischensteht, abzutippen, sondern die in Tabelle 1 aufgeführten Tasten zu drücken. Auf Ihrem Bildschirm erhalten Sie dann wieder die entsprechenden Grafikzeichen.

9. Alle Grafikzeichen werden ebenfalls ersetzt durch unterstrichene oder überstrichene Großbuchstaben.

Unterstrichene Buchstaben bedeuten, daß Sie die <SHIFT>-Taste und den angegebenen Buchstaben drücken müssen, überstrichene jedoch die Commodore-Taste (<CBM>) oder (<C= >) mit dem Buchstaben.

Auch hier erhalten Sie am Bildschirm das entsprechende Grafikzeichen und nicht etwa das im Listing erkennbare Zeichen.

Die Leerzeichen zwischen den einzelnen Basic-Befehlen können beim Abtippen entfallen (ohne Einfluß auf die Checksumme zu nehmen). Dies ist besonders bei speicherkritischen Programmen wichtig. Ebenso müssen Zeilen, die mehr als 80 Zeichen pro Zeile enthalten, mit den bekannten Abkürzungen für die Basic-Befehle (siehe auch das Handbuch zum C64, Anhang D, Seite 130) eingegeben werden.

Sie können die Programme auch weiterhin ohne den Checksummer eintippen.

(F. Lonczewski/gk)

Hinweis: {13 SPACE} bedeutet 13mal die Leertaste drücken

```
9 REM ********************************
10 PRINT"{CLR,11SPACE,RVSON}CHECKSUMMER 64
   V3{RVOFF}"
11 PRINT"{2DOWN,9SPACE}EINEN MOMENT, BITTE
   ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
   A+1:NEXT I
13 IF PS<>5802 THEN PRINT"PRUEFSUMMENFEHLE
   R IN ZEILEN 20-22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
   A:POKE I,A:PS=PS+A+1:NEXT I
15 IF PS<>16267 THEN PRINT"PRUEFSUMMENFEHL
   ER IN ZEILEN 22-30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT"{4DOWN,9SPACE}CHECKSUMMER AKTIVIE
   RT."
18 PRINT"{2DOWN}AUSSCHALTEN : POKE1,55"
19 PRINT"{DOWN}ANSCHALTEN{2SPACE}: POKE1,5
   3":NEW
20 DATA 169,0,133,254,162,1,189,93,3,133,2
   55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
   5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
   0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
   ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
   202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
   ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
   9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
   32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
   9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
   29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
```


Listing. Der Checksummer 64 V3 erkennt auch Vertauschungen von Zahlen

CTRL steht für Control-Taste, so bedeutet {CTRL+A}, daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{DOWN}	Taste neben rechtem Shift, Cursor unten
{UP}	Shift-Taste & Taste neben rechtem Shift; Cursor hoch
{CLR}	Shift-Taste & 2. Taste ganz rechts oben
{INST}	Shift-Taste & Taste ganz rechts oben
{HOME}	2. Taste von ganz rechts oben
{DEL}	Taste ganz rechts oben
{RIGHT}	Taste ganz rechts unten
{LEFT}	Shift-Taste & Taste unten rechts
{SPACE}	Leertaste
{SHFT-SPCE}	Shift-Taste & Leertaste
{F1} bis {F8}	Funktionstasten
{RETURN}	Shift-Taste & Return
{BLACK}	Control-Taste & 1
{WHITE}	Control-Taste & 2
{RED}	Control-Taste & 3
{CYAN}	Control-Taste & 4
{PURPLE}	Control-Taste & 5
{GREEN}	Control-Taste & 6
{BLUE}	Control-Taste & 7
{YELLOW}	Control-Taste & 8
{RVSON}	Control-Taste & 9
{RVOFF}	Control-Taste & 0
{ORANGE}	Commodore-Taste & 1
{BROWN}	Commodore-Taste & 2
{LIG.RED}	Commodore-Taste & 3
{GREY 1}	Commodore-Taste & 4
{GREY 2}	Commodore-Taste & 5
{LIG.GREEN}	Commodore-Taste & 6
{LIG.BLUE}	Commodore-Taste & 7
{GREY 3}	Commodore-Taste & 8

Tabelle 1. Die Steuerbefehle in den Listings

Ergänzen Sie jetzt Ihre 64'er-Sammlung

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle Ausgaben von 64'er? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?

Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der Ausgaben 01 bis 12/85.

Und so kommen Sie schnell an die noch lieferbaren Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

Aktuell

Stichwort	Titel	Seite	Ausgabe
Allgemeines	Commodore Gestern Heute Morgen	10	01/85
Computer	Amiga — Der neue Supercomputer	8	09/85
Interview	Interview mit David Crane (Game Designer)	146	06/85
Lernen	Schule braucht Computer (VAM-Computer)	9	06/85
Messen	International Chaos Communication Congress	15	03/85
	Heiße Messe in der Wüste: CES	8	03/85
	Hannover-Messe '85	8	06/85
	Hannover-Messe '85	8	07/85
	Chicago im Zeichen der CES	8	08/85
	Aktuelles von der C'85 in Köln	15	08/85
	Btx Total (Internationale Funkausstellung)	8	10/85
	PCW-Computermesse in London	8	11/85
	Neues von der Commodore-Fachausstellung 1985	8	12/85
Recht	Die neue Abmahnmasche — Vorsicht bei Programmangeboten	8	05/85
	Die Ex-Knacker — wo sind sie geblieben?	27	08/85
	Interview mit Raubkopierern (Section 8)	28	08/85
	Schützer kontra Knacki's	23	08/85
	Raub-Talkshow	12	08/85
	Das Urheberrechtsgesetz und Gedanken zu seiner Anwendung	21	08/85
	Änderung des Urheberrechtsgesetzes	162	09/85

Buchbesprechungen

Stichwort	Titel	Seite	Ausgabe
Anfänger	Goldmann Computer Compact	87	03/85
	Basic-Wegweiser für den C 64	86	05/85
	Alles über den C 64, Sachbuchreihe, Band 1	115	06/85
	Lehrspielzeug Computer: C 64/VC 20	112	11/85
	C 64 Computerhandbuch	171	11/85
	Einführungskurs: Commodore 64	144	12/85
Anwendung	Dienstprogramme VC 20, C 64 und SX	86	05/85
	Spaß an Mathe mit dem Commodore 64	88	07/85
	Mathe für die Oberstufe mit dem C 64	88	07/85
	Mathematische Routinen VC 20, Elektrotechnik/Elektronik	112	11/85
	Commodore 64-Listings, Band 2: Dateiverwaltung, Schule, Hobby	112	11/85
	Das Trainingsbuch zum Datamat	144	12/85
C 128	Bücher zum C 128	22	10/85
DFÜ	Das Mailbox-Jahrbuch: Nutz die Netze	112	11/85
Grafik	Grafik auf dem Commodore 64 (+Fehlert. 9/85)	86	05/85
	Einführung in CAD mit dem Commodore 64	128	06/85
	Grafik & Musik auf dem Commodore 64	88	07/85
	Verschiedene Grafikbücher zum C 64	115	08/85
Programmieren	Von Basic zu Assembler: Das Commodore-Buch, Band 4	115	06/85
	64 Intern	115	06/85
	Das Interface Age System-Handbuch zum C 64	115	06/85
	Das C 64 Buch, Band 5: Simons Basic Leitfaden	144	12/85
	Basicode	144	12/85
	Noch mehr Tips und Tricks zum 64er	144	12/85
Speichern	Das Kassettenbuch zum C 64 und VC 20	87	03/85
	Die Floppy 1541 (M&T)	88	07/85
Spiele	Rombachs C 64 Spielführer	87	03/85
	Commodore 64-Listings, Band 1, Spiele	112	11/85
	35 ausgesuchte Spiele für Ihren Commodore 64	171	1/85

64'er Extra

Stichwort	Titel	Seite	Ausgabe
Prozessor	Befehlssatz des 6502/6510-Prozessors	84	09/85
Grafik	Die Videochip-Register des C 64	92	10/85
Sound	Der SID-Chip, seine Register und Programmierung	92	11/85
Speicher	Die Speicherbelegung des C 64	96	12/85

Abenteuerlösungen

Stichwort	Titel	Seite	Ausgabe
Lösungen	Dallas-Quest Lösung	90	01/85
	Guncho Krill-Enchanter ist gelöst	44	03/85
	Infocom-Geheimnisse gelüftet?	49	05/85
	Des Rätsels Lösung: Amazon	145	06/85
	Activision-Adventures entschleiert (Mindshadow, Tracer Sanction)	36	12/85
	Eureka! — ich hab's!	37	12/85
	Lösungen zu Hitchhiker's Guide und Sorcerer	39	12/85

Spiele-Tests

Stichwort	Titel	Seite	Ausgabe
007	James Bond — A View to a Kill	156	09/85
Abenteuer	Abenteuerpaket 1	48	08/85
	Shadowfire	146	09/85
	The Quest — mit C 64 auf Suche nach Drachen	47	01/85
Action	Hexenküche	50	07/85
	Master of the Lamps	48	07/85
	Rescue on Fractalus	158	10/85
	Stellar 7	49	08/85
Construction Set	Mail Order Monsters	49	08/85
	Racing Destruction Set	50	08/85
Geschicklichkeit	Australopedicus Robustus	50	08/85
	Boulder Dash II	159	10/85
	Crystal Castles	50	07/85
	Gribbly's Day out	148	09/85
	Rock'n Bolt	48	08/85
	Thing on a Spring	159	10/85
	Tom + Zaga	48	01/85
Pseudo-Adventures	Roland's Rat Race	49	08/85
	Fourth Protocol und Frankie g.t.H.	162	11/85
Renner	Die Renner 1985: Meistverkaufte Spiele	34	12/85
Schach	Viermal Schachmatt: Verschiedene Schachprogramme	32	12/85
Simulation	Elite	148	09/85
	Jump Jet	148	09/85
	Super Huey Hubschraubersimulator	49	07/85
Sport	Boxspiele: Frank Bruno's B. + Barry McGuigan Champions. B.	49	12/85
	Handkantenschlag per Joystick: Karateka + Exploding Fist	165	11/85
	Nick Faldo Plays the Open (Golf)	159	10/85
	Rallye Speedway	49	07/85
	Slapshot (Eishockey)	50	07/85
	Summer Games II	146	09/85
	World Series Baseball	49	07/85
Diverses	New York City und Air Support	145	06/85

Hardware-Tips und Bauanleitungen

Stichwort	Titel	Seite	Ausgabe
Audio/Video	Mit 5 Mark zu neuen Dimensionen (Stereoanlage am C 64)	34	05/85
	Ein Monitor ist genug (RGB+Composite an C 128)	16	10/85
C 16	Alte Datasette am C 16	31	04/85
	Alter Joystick am C 16	35	05/85
Eingabegeräte	Der Hexer — Zusatztastatur für den MSE	48	10/85
EPROM	EPROMs im Expansion-Port	46	10/85
	EPROM Trans — Die Super-Erweiterung	42	10/85
	Das 64'er EPROM-Programmiergerät, Teil 1	44	12/85
Floppy/Datasette	Diskettenlaufwerk 1541 selbst justiert	32	10/85
	Die Datasette streikt nie wieder (Anpassung des Tonkopfs)	34	10/85
IEC-Bus	Auf zu neuen Welten: IEC-Bus im Selbstbau (+Fehlerteufel 10/85)	44	07/85
Joystick	Joystick im Selbstbau	33	03/85
	Dauerfeuer-Adapter	46	08/85
RS232/V.24	Das 30-Mark-Interface (Selbstbau RS232)	29	03/85
	Genau betrachtet: Die RS232/V.24-Schnittstelle	80	05/85
Diverses	Userport-Display	36	05/85
	Reset-Taster für alle Fälle (+Fehlert. 9/85)	130	06/85
	Aus eins mach vier (absturzfreie Betriebssystemumschaltung)	41	07/85

Hardware-Grundlagen

Stichwort	Titel	Seite	Ausgabe
Computer	Was bringt der C 128?	28	11/85
Drucker	Welcher Drucker ist der Richtige? (Grundlagen)	15	05/85
	Hammerwerke — wie funktionieren Typenraddrucker	32	06/85
	Die Alternativen: Thermo-, Tintenstrahldrucker + Plotter	24	07/85
Eingabegeräte	Versteht Sie Ihr Computer? (Wie funktionieren Eingabegeräte)	44	09/85
Floppy	Floppy oder Datasette?	129	06/85
Monitore	Wie funktionieren sie, was ist beim Kauf zu beachten?	16	12/85
	Das Kabel zum Monitor: Welche Normen gibt es?	28	12/85
Peripherie	Grafikeingabegerät: Wie funktionieren sie?	30	08/85

Hardware-Tests

Stichwort	Titel	Seite	Ausgabe
Computer	Generationswechsel: Test C 16	16	01/85
	Erster ausführlicher Test C 128 PC (Teil 1)	16	06/85
	Erster ausführlicher Test C 128, PC (Teil 2)	17	07/85
DFÜ	Marktübersicht Modems & Akustikkoppler	32	07/85
Drucker	Vergleich: Drucker unter 700 Mark (Tests und Marktübersicht)	18	05/85
	Tests und Marktübersicht Typenraddrucker	35	06/85
	Test: Brother EP 44	27	07/85
	Brother TC-600	118	08/85
	Riteman C+	133	09/85
	Panasonic KX-P1091	134	09/85
	Star SG 10C	132	09/85
	Melchers CP-80X — wie hätten Sie's denn gern?	25	10/85
	Geheimtip: Der RFI DP 165	24	10/85
	Epson GX 80 — einer für alle	26	10/85
	MPS 803 — ein Drucker für alle Gelegenheiten?	40	1/85
	Epson JX-80 das vielfarbige Druck-Genie	38	11/85
	Epson FX-85 neue Referenz	42	11/85
	SP 1000 VC — Superstar mit Haken	41	11/85
	Der NEC-P2 — das fernöstliche Wunder	159	12/85
	DMPG9 — eine solide Sache	162	12/85
	Das Doppelleben des Joystick-Ports: 10er-Tastaturen	50	09/85
	Joysticks: Test und Marktübersicht (+Fehlerteufel 12/85)	19	11/85
	Es geht auch anders: Lightpens und Trackballs	22	11/85
EPROMer	Frisch gebrannt ist halb gespeichert (EPROM-Programmiergeräte im Test)	39	07/85
	QuickByte II — das Kraftpaket	14	10/85
Floppy/Datasette	Turbo-Floppies, zweite Generation: Speeddos plus + Prologic DOS	28	10/85
	Das große Rennen: Schnelle Bandlaufwerke	37	10/85
	Professionelle Floppylaufwerke für den C 64 (IEC-Floppies)	30	10/85
	Gut gekauft ist halb gespeichert (Marktübersicht Disketten)	38	10/85
Grafik	Die Videowerkstatt (Digitizer-Test)	32	05/85
	Digitalbilder m.d. C 64: PrintTechnik Digitizer	24	01/85
Interface	Hardware-Interface ganz weich: Test EC 64	23	01/85
	Gute Connections — Übersicht Schnittstellen	21	03/85
	Card/Print +6 — Das Allround-Interface	20	03/85
	Das Wiesemann-Centronics-Interface	18	03/85
	Erst ein IEC-Bus öffnet Tür und Tor (+Fehlert.4/6-85)	24	03/85
Monitore	Marktübersicht: Monochrome Monitore	30	12/85
Musik	Trommelwirbel: Test Digital Drums	45	08/85
	Die Musikhardware zum C 64	17	09/85
Roboter	Roboter selbst gebaut (Fischertechnik)	167	10/85
Scanner	So lernt Ihr Drucker lesen	30	06/85
Speicher	Speichertuning VC 20: Test 64 KByte Karte	26	01/85
Steuern	Flottes Türmchen: MEA-Interface	116	08/85

Kurse

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler ist keine Alchimie, Teil 5	142	01/85
	Assembler ist keine Alchimie, Teil 7	124	03/85
	Assembler ist keine Alchimie, Teil 9	138	05/85
	Assembler ist keine Alchimie, Teil 10	127	07/85
	Assembler ist keine Alchimie, Teil 11	126	08/85
	Assembler ist keine Alchimie, Teil 12	109	09/85
	Assembler ist keine Alchimie, Teil 13 (Schluß)	143	10/85
C 128	Entdeckungsreise duch den C 128	42	12/85
Effektives Programieren	Müllabfuhr im Computer: Garbage Collection, Teil 1	122	01/85
	Finden mit System, eine neuartige Suchmethode, Teil 3	148	03/85
	Sortieren mit dem Computer, Teil 2	159	05/85
	Sortieren mit dem Computer, Teil 3	124	06/85
	Sortieren mit dem Computer, Teil 4	138	08/85
	Sortieren mit dem Computer, Teil 5	124	09/85
	Sortieren mit dem Computer, Teil 6 (Schluß)	150	12/85
Extern	C 64 extern — Der Weg nach draußen, Teil 1	144	08/85
	C 64 extern — Der Weg nach draußen, Teil 2	122	09/85
	C 64 extern — Der Weg nach draußen, Teil 3 (Schluß)	129	10/85
Floppy	In die Geheimnisse der Floppy eingetaucht, Teil 4	148	01/85
	In die Geheimnise der Floppy eingetaucht, Teil 5	130	03/85
	In die Geheimnise der Floppy eingetaucht, Teil 6	145	05/85
	In die Geheimnisse der Floppy eingetaucht, Teil 7 (Schluß)	116	06/85
	Directory-Manipulationen I	140	06/85
Floppy	Directory-Manipulationen II	163	10/85
Grafik	Hires 3 — 15 neue Basic-Befehle, Teil 2	136	03/85
	Hires 3 — Grafikkurs-Anwendung, Teil 3 (Schluß)	152	08/85
	Sprites ohne Geheimnisse	40	08/85
	Streifzüge durch die Grafikwelt, Teil 1	106	09/85
	Streifzüge durch die Grafikwelt, Teil 2	149	11/85
Logeleien	Logeleien, Teil 1	143	07/85
	Logeleien, Teil 2	136	08/85
	Logeleien, Teil 3 (Schluß)	115	09/85
Musik	Dem Klang auf der Spur, Teil 2	136	01/85
	Dem Klang auf der Spur, Teil 4	131	04/85
	Dem Klang auf der Spur, Teil 5	152	05/85
	Dem Klang auf der Spur, Teil 7	132	07/85
	Dem Klang auf der Spur, Teil 8	133	08/85
	Dem Klang auf der Spur, Teil 9	126	10/85
	Dem Klang auf der Spur, Teil 10 (Schluß)	157	11/85
Speicher	Memory Map mit Wandervorschlägen, Teil 3	126	01/85
	Memory Map mit Wandervorschlägen, Teil 5	144	03/85
	Memory Map mit Wandervorschlägen, Teil 7	120	06/85
	Memory Map mit Wandervorschlägen, Teil 8	140	07/85
	Memory Map mit Wandervorschlägen, Teil 9	129	08/85
	Memory Map mit Wandervorschlägen, Teil 10	112	09/85
	Memory Map mit Wandervorschlägen, Teil 11	133	10/85
	Memory Map mit Wandervorschlägen, Teil 12	145	11/85
	Memory Map mit Wandervorschlägen, Teil 13	146	12/85
Sprachen	Basic ist out — es lebe Forth	43	01/85
VC 20	Der gläserne VC 20, Teil 4	130	01/85
	Der gläserne VC 20, Teil 6 (Schluß)	155	03/85

Software-Tips

Stichwort	Titel	Seite	Ausgabe
C 128	Erste Fragen und Antworten zum C 128	14	09/85
	Fragen und Antworten zum 128er	20	10/85
	Fragen und Antworten zum 128er	40	12/85
Drucker	Der MPS 802 lernt Deutsch	30	05/85
	Centronics-Interface für jeden Bedarf	78	07/85
Textverarbeitung	Software Corner — professionelle Programme richtig eingesetzt (Vizawrite-Tips)	174	12/85
Tips & Tricks	Autoboot beim C 64	86	03/85
	Verbindungsfreundlich (Parallelschnittstelle des VC 20)	91	03/85
	Undefinierte Opcodes des 6502	84	03/85
	Durch POKEs zum Erfolg (Spiele-POKEs)	83	03/85
	Tips und Erweiterungen zu Hi-Eddi und Simons Basic	88	03/85
	Basic-Befehle im Griff	79	05/85
	Durch POKEs zum Erfolg: Spiele-POKEs	78	.06/85
	Formatierte Eingabe	148	06/85
	Hi-Text (Text in Hires)	70	08/85
	Verbotene Variablen	66	09/85
	Verschiedene Routinen für Anfänger und Profis (+Fehlerteufel 12/85)	88	11/85
	Der Trick mit dem Joystick (Joystickabfrage)	24	11/85
	Verschiedene Tips für Anfänger und Fortgeschrittene	106	12/85

Software-Grundlagen

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler? Assembler! (Einführung)	32	01/85
	Assembler-Bedienung leicht gemacht, Teil 1	169	12/85
DFÜ	Der erste Kontakt mit DFÜ	40	06/85
	Die Netze der Post: Btx, Datex-P, Telebox	46	06/85
	DFÜ — Was ist das?	44	06/85
	Mailbox für Anfänger	30	07/85

Stichwort	Titel	Seite	Ausgabe
Datei	Die wichtigsten Begriffe der Dateiverwaltung	42	05/85
	Dateiverwaltung ist nicht gleich Datenbank	44	05/85
	Dateiverwaltung: Was Sie beim Kauf beachten sollten	40	05/85
Drucker	Hardcopy leicht gemacht (wie programmiert man Hardcopies)	34	09/85
EPROM	Wie sage ich es meinem EPROM? (EPROM-Grundlagen)	35	07/85
Funktionen	Funktionen für Anfänger	164	05/85
Lernen	Besser lernen mit dem Computer	166	10/85
Musik	Klangprogrammierung ohne Ballast	19	09/85
Spiele	Taktik- und Strategiespiele	46	03/85
	Play by Mail und Play by Modem	153	09/85
Sprachen	Sprachen für Computer, Teil 2	46	05/85
Textverarbeitung	Von der Schreibmaschine zum Textsystem	34	03/85

Listings zum Abtippen

Stichwort	Titel	Seite	Ausgabe
Anwendung	Der C 64 als Handballtrainer (AdM)	52	01/85
	Ligatab — ohne Organisation kein Tor (LdM)	50	03/85
	Gut Ziel mit dem C 64 — Schützenvereinsergebnisse (AdM)	52	03/85
	Weißt du, wieviel Sternlein stehen (Sternkarte) (AdM) (+ Fehlert. 6/85)	52	05/85
	Haushaltsbuchführung (AdM)	52	07/85
	Netzwerkanalyse: Ein Programm für Hobby-elektroniker (AdM)	52	08/85
	Prüfungsfragen (AdM)	52	09/85
	Fit in Latein mit dem C 64 (AdM)	52	10/85
	Lyrik-Maschine (AdM)	52	11/85
	Hypra-Platos (LdM)	50	11/85
	Der Chemie-Assistent (AdM)	52	12/85
	SMON Teil 3: Ohne gutes Werkz. geht es nicht	69	01/85
	Hypra-Ass (LdM)	51	07/85
	Neues vom SMON (+ Fehlerteufel 11/85)	87	10/85
	Reassembler zu Hypra-Ass (+ Fehlerteufel 12/85)	97	11/85
	Ergänzungen zu Hypra-Ass (bedingte Verzweigungen)	96	11/85
	Tips & Tricks zum SMON (inklusive Diskmonitor)	100	12/85
Bildschirmseite	Auflösung Wettbewerb Bildschirmseite: Drei Top-Programme	158	09/85
DFÜ	Terminalprogramm der Spitzenklasse (+ Fehlerteufel 10/85)	149	07/85
Datei	SMU — Der Maskengenerator (LdM)	50	12/85
Drucker	Hi-Eddi-Druckerroutinen	69	06/85
	C 64 Schreiberling — Drucken wie gemalt	54	10/85
	Koalabilder Farbhardcopy auf Epson JX-80	39	11/85
Einzeiler	Die nächsten 14 aus d. Einzeilerwettbewerb	157	01/85
Floppy	Hypra-Load mal 4 (+ Fehlerteufel 3/85)	82	01/85
	Diskettenmonitor	83	08/85
	Disk-Designer	70	09/85
	Herzoperation (Hypra-Load + Hypra-Ass + DOS5.1 + Centronics)	104	11/85
Grafik	Vier Pseudo-VICs mit 32 Sprites	76	01/85
	Hi-Eddi: Zeichen- und Malproggramm (LdM)	50	01/85
	Elektrotechnisches Zeichnen mit dem VC 20	71	03/85
	Mini-Grafik VC 20, Grafikhilfe	69	05/85
	Trickfilm mit dem C 64: Bewegte 3D-Grafik (LdM) (+ Fehlerteufel 6/85)	51	05/85
	Kurvenplotten mit Hardcopy auf dem C 16	68	06/85
	Doppelte Grafikauflösung für C 128	33	11/85
	Bilder aus einer anderen Dimension (Apfelmännchen)	80	11/85
Intelligenz	VIC — das intelligente Programm (Wettbewerbssieger)	173	05/85
Musik	Sound Machine (+ Fehlerteufel 10/85)	23	09/85
	Sound Master (Basic-Erweiterung)	31	09/85
Spiele	6510 — Die Suche nach der Prozessor	70	05/85
	Samurai (Strategiespiel)	72	06/85
	Schach dem C 64: Schachprogramm zum Abtippen	72	08/85
	Spielen auf zwei Bildschirmen: Zeichensatzscrolling (LdM)	51	09/85
	Pac-Man unter der Lupe	76	10/85
	Block Out	84	11/85
	Seekrieg per Telefon (Schiffe versenken per Modem)	82	12/85
Spielehilfe	Die Scroll-Maschine — D. Fenster zur Spielewelt (LdM) (+ Fehlert. 11/85)	52	06/85
Sprachen	Tiny Forth Compiler (LdM) (+ Fehlert. 9/85)	51	08/85
Textverarbeitung	Hypra-Text (LdM) (+ Fehlerteufel 11/85)	50	10/85
	Drucksache — Hypra-Text, Teil 2	71	11/85
Tips & Tricks	Große Buchstaben	89	01/85
	Restore für Unterprogramme	90	01/85
Tips & Tricks	Parameterübergabe an Maschinenspracheprogramme	88	01/85
	Cursorsteuerung leicht gemacht	86	02/85
	22 Read Error — Theorie und Praxis	41	03/85
	Floppy-Lister (+ Fehlerteufel 4/85)	82	03/85
	Longscreen beim VC 20	83	05/85
	C 16: Help und Trace verbessert	84	05/85
	Ordnung ist das halbe Leben (Directory-Sorter)	77	05/85
	Dokumentationshilfe, Cross-Referenz-Liste C 64 (Wettbewerb)	155	06/85
	Prost mit dem C 64: Gerätesteuerung über Userport (+ Fehlerteufel 9/85)	76	06/85
	Fenster-Befehle für den C 16	84	07/85
	Elektronische Merkzettel	83	07/85
	File-Compactor	82	07/85
	REM-Killer (+ Fehlerteufel 9/85)	75	07/85
	Basic-Start-Generator	74	07/85
	Komfortable Ein-/Ausgaberoutine	77	07/85
	Bildschirmmasken leicht erstellt	86	08/85
	Der Bitmap-Compander (HiRes-Bilder komprimieren)	81	08/85
	Hypra-Save	79	08/85
	'Procedure' — oder der C 64 kann lernen	78	08/85
	Aufgewickelt — Listingscrolling für VC 20	63	09/85
	Programmgenerator für den C 64	86	10/85
	Cross-Ref optimiert	83	10/85
	Spieletrainer: Spritekill	86	11/85
	Tipp-Utility	99	12/85
	Der EPROM-Automat (wie man Module macht)	90	12/85
	80-Zeichen-Grafik für den C 128	78	12/85
	Hyper Screen (Sprites auf dem Bildschirmrand)	76	12/85
Transfer	Der C 64 als PET: PET-Simulator	87	01/85
Unterprogramme	Formatierte Eingabe	156	01/85

Software-Tests

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler im Test Teil 1	34	01/85
Basic-Erweiterung	GBasic — Alles drin	28	01/85
	Macro-Basic: Die Unterprogramm-Bibliothek	137	06/85
	Darf es etwas mehr sein? — Test Business-Basic	120	08/85
	Das Intellectool	138	09/85
	Formel 64: Das Multitalent	158	12/85
DFÜ	Terminalprogramme: Übersicht	42	06/85
Datei	Vergleichstest — 7 Dateiverwaltungen auf einen Blick	118	07/85
	Aufgeräumt mit Mainfile II	157	10/85
Grafik	Malen auf dem Bildschirm (Malprogramme)	34	08/85
	Grafikprogramme auf einen Blick: Marktübersicht	38	08/85
	Vergleichstest: Grafik-Erweiterungen	37	09/85
Lernen	Softlearning — die weiche Welle des Lernens	40	01/85
	Vokabeltraining mit dem Computer	39	03/85
	Marktübersicht: Lernsoftware	168	10/85
Musik	Musik für den C 64: Übersicht Musiksoftware	26	09/85
	The Music System — Zwei auf einen Schlag	164	12/85
Sprachen	Logo — die Sprache für Einsteiger	135	05/85
	Der Ada Trainingskurs auf dem C 64	129	05/85
	Promal — die neue Sprache für Profis?	124	07/85
	Forth-wärts mit M&T-Forth 64	126	07/85
	Was leistet Pilot?	121	08/85
	Pascal für Profis (Profi-Pascal)	122	08/85
	Super-Forth 64	144	09/85
	C — die professionelle Programmiersprache für den C 64	140	09/85
	Basic 7.0 — Das Superbasic des C 128	18	10/85
	Comal 80 — die universelle Programmiersprache	151	10/85
	Turbo-Pascal auf dem C 128	30	11/85

Stichwort	Titel	Seite	Ausgabe
Textverarbeitung	Homeword - Textverarbeitung zu Hause	36	03/85
	Totl-Text — Flexibilität ist Trumpf	38	03/85
	Protext — Textprofi mit 80 Zeichen	133	05/85
	Textomat Plus kontra Vizawrite	132	06/85
	Der Preishammer (Test: StarTexter)	135	09/85
	Paperclip — ausdrücklich gut	44	11/85

So machen's andere

Stichwort	Titel	Seite	Ausgabe
Semmeln	Semmelservice mit dem C 64	147	06/85
Sport	Commodore Sportservice: Heimcomputer zur Turnierauswertung	157	07/85
Hilfe	Computer für Behinderte	182	12/85

MSE – Abtippen sicher und leicht gemacht

Ähnlich wie der Checksummer ist auch der MSE ein leicht zu bedienendes Hilfsmittel bei der Eingabe von Listings, diesmal jedoch bei reinen Maschinensprache-Programmen.

Im Gegensatz zum »Checksummer« aber ist die Eingabe nicht ohne den MSE möglich. Der MSE verringert die Tipparbeit um ein Drittel und schließt Fehleingaben vollkommen aus. Außerdem können Sie die Werte blind eingeben, ohne andauernd auf den Bildschirm schauen zu müssen. Dies wird durch akustische Meldungen realisiert.

MSE ist ein Maschinensprach-Editor, mit dem ein Vertippen ausgeschlossen ist. Eine abgetippte Zeile wird nur angenommen, wenn sie richtig ist. Eine Checksumme am Ende jeder Zeile prüft, ob die richtigen Werte in der richtigen Zeile an der richtigen Stelle stehen. Wenn nicht, ertönt ein Warnsignal, und man beseitigt den Fehler.

War die Zeile korrekt, erklingt ein Gong, und die nächste Zeilennummer wird ausgegeben. Damit ist also auch »blindes« Eintippen möglich; Sie können sich voll auf den Text konzentrieren.

So arbeitet man mit MSE

Laden und starten Sie MSE. Zuerst werden der Programmname und die Start- und Endadresse erfragt. **Diese Angaben entnehmen Sie dem Kopf des jeweiligen abgedruckten Listings.** Der MSE meldet sich dann mit der Zeilennummer der ersten Zeile.

Wenn Sie die Zeile richtig eingegeben haben, erscheint die nächste Zeilennummer und so weiter bis zum Ende. Zum Schluß wird das fertige Programm mit <CTRL+S> auf Diskette oder Kassette gespeichert. Dazu sind keine weiteren Angaben mehr erforderlich. Das Programm kann dann ganz normal wieder geladen und gestartet werden. Wenn Sie nicht alles auf einmal tippen wollen, können Sie jederzeit unterbrechen und den eingetippten Teil mit <CTRL+S> speichern. Wollen Sie weiterarbeiten, laden und starten Sie MSE wieder.

Geben Sie auf die Frage nach der Startadresse aber jetzt <L> ein, um Ihr Teilprogramm zu laden. Jetzt können Sie mit <CTRL+N> die Adresse eingeben, an der Sie weitertippen müssen. Wenn Sie sich nicht gemerkt haben, wie weit Sie gekommen sind, geben Sie nach dem Laden <CTRL+M> ein.

Auf die Frage nach der Startadresse antworten Sie mit der Anfangsadresse, die links in der Kopfzeile auf dem Bildschirm steht. Nun wird Ihr Programm aufgelistet. Mit <SPACE> wird das Listen fortgesetzt, mit <STOP> abgebrochen. Das Ende Ihres Programmteils erkennen Sie sehr einfach daran, daß nur noch der Wert »AA« in der Zeile steht. Die Adresse dieser Zeile müssen Sie anschließend mit <CTRL+N> eingeben. Das Programm ist nur mit <RUN/STOP+RESTORE> zu verlassen. Speichern Sie aber vorher unbedingt immer Ihren Text.

Hinweise zum Abtippen

Vor dem Abtippen oder späteren Wiederladen des MSE-Laders müssen Sie unbedingt folgende Zeile eingeben:
POKE 43,1: POKE 44,32: POKE 8192,0: NEW

Den MSE-Lader brauchen Sie nur einmal. Nach erfolgreichem Abtippen und Starten mit RUN geht der Lader verloren, und es wird das endgültige Programm MSE V1.0 erzeugt. So gehen Sie vor:

Starten Sie das Programm mit RUN. Fehlerhafte Zeilen werden angezeigt und müssen korrigiert werden, bis der Lader zum »READY« durchläuft. Jetzt müssen Sie das fertige MSE-Programm speichern. Dazu brauchen Sie nur <RETURN> zu drücken, weil die erforderlichen Angaben schon auf dem Bildschirm stehen. (Kassettenbesitzer müssen in Zeile 343 die letzte Zahl in »1« abändern.) Ab jetzt können Sie »MSE V1.0« direkt, also ohne den DATA-Lader, benutzen. MSE V1.0 wird ganz normal mit »,8« geladen (keine POKEs notwendig).

(N. Mann/D. Weineck/gk)

MSE-Befehle:

<DEL>	löscht die letzte Eingabe.
<CTRL+S>	speichert das eingetippte Programm ab.
<L> oder <CTRL+L>	lädt ein Programm. Start- und Endadresse werden automatisch ermittelt.
<CTRL+M>	listet den Speicherinhalt. Abbruch mit STOP-Taste, weiter mit Leertaste.
<CTRL+N>	erlaubt die Eingabe einer neuen Adresse zum Weitertippen.
<CTRL+P>	gibt ein MSE-Listing auf dem Drucker aus.

```
100 REM ****************************         <091>
110 REM *                          *         <159>
120 REM *         M S E   LADER    *         <206>
130 REM *                          *         <179>
220 REM ****************************         <211>
230 REM                                      <036>
240 DIM H(75): FOR I=0 TO 9                  <113>
250 H(48+I)=I: H(65+I)=I+10:NEXT             <041>
260 FOR I=2048 TO 3755 : READ A$             <198>
270 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))   <199>
280 D=H(H)*16+H(L):S=S+D:POKE I,D            <219>
290 A=A+1:IF A<20 THEN NEXT:A=-1             <141>
300 PRINT " ZEILE:";1000+Z;                  <011>
310 READ V :Z=Z+1:IF V=S THEN 330            <218>
320 PRINT"PRUEFSUMMENFEHLER !":STOP          <138>
330 IF A<0 THEN 341                          <221>
340 S=0:A=0:PRINT:NEXT                       <046>
341 PRINT"{CLR}P0̲43,1:P0̲44,8:P0̲45,172:P0̲46
    ,14                                      <010>
342 POKE 631,19:POKE 632,13:POKE 633,13:PO
    KE 198,3                                 <249>
343 PRINT"{3DOWN}SAVE"CHR$(34)"MSE V1.0"CH
    R$(34)",8                                <171>
344 END                                      <092>
1000 DATA 00,0B,08,0A,00,9E,32,30,36,31,00
     ,00,00,A2,08,A9,36,85,A4,A9, 1247       <119>
1001 DATA 08,85,A5,A9,00,85,A6,A9,B0,85,A7
     ,A0,00,B1,A4,91,A6,C8,D0,F9, 2888       <054>
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01
     ,4C,00,B0,20,D1,B1,A9,06,8D, 2787       <144>
1003 DATA 21,D0,A9,03,8D,20,D0,8D,86,02,A0
     ,B3,A9,74,20,FF,B1,A0,B3,A9, 2667       <237>
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01
     ,02,C8,C9,0D,D0,F5,88,F0,D2, 2912       <217>
1005 DATA C0,0F,90,02,A0,0E,8C,00,02,20,EA
     ,B1,A0,B3,A9,CF,20,FF,B1,20, 2323       <013>
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB
     ,85,61,20,A7,B4,D0,20,A0,B3, 2864       <199>
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20
     ,8E,B4,85,5F,20,A7,B4,D0,0A, 2624       <091>
```

Der MSE zum bequemen Abtippen von Maschinenprogrammen

```
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20
     ,43,B3,4C,3A,B0,A9,AA,A0,00, 2379     <167>
1009 DATA 91,FB,E6,FB,D0,02,E6,FC,20,3F,B2
     ,90,EF,4C,FB,B4,A2,02,86,58, 3118     <152>
1010 DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0
     ,FB,C9,30,90,0C,C9,47,B0,08, 2970     <231>
1011 DATA C9,3A,90,0B,C9,41,B0,07,C9,14,D0
     ,0F,4C,0B,B1,20,D2,FF,A6,58, 2322     <121>
1012 DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0
     ,26,C9,0C,D0,03,4C,0B,B6,C9, 2685     <057>
1013 DATA 13,D0,03,4C,8B,B5,C9,0D,D0,03,4C
     ,BA,B4,C9,10,D0,03,4C,68,B5, 2282     <225>
1014 DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C
     ,92,B0,A5,F9,20,02,B1,0A,0A, 2132     <208>
1015 DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9
     ,60,C9,3A,90,02,69,08,29,0F, 1950     <092>
1016 DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02
     ,B0,06,20,D2,FF,4C,8E,B0,C6, 2509     <188>
1017 DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA
     ,84,57,68,68,4C,8B,B1,A6,D3, 2891     <197>
1018 DATA E0,08,B0,03,4C,92,B0,20,D2,FF,A6
     ,58,E0,02,90,09,C6,59,20,D2, 2468     <049>
1019 DATA FF,C6,58,D0,F9,4C,8E,B0,48,4A,4A
     ,4A,4A,20,59,B1,68,29,0F,C9, 2419     <035>
1020 DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2
     ,FC,9A,20,D1,B1,20,48,B2,20, 2261     <073>
1021 DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5
     ,FB,20,4E,B1,20,ED,B1,A9,3A, 2860     <148>
1022 DATA A0,20,20,F2,B1,A9,00,85,59,20,8E
     ,B0,20,ED,B1,A4,59,20,EF,B0, 2530     <233>
1023 DATA 91,FB,C8,84,59,C0,08,90,EC,20,10
     ,B2,A9,12,20,D2,FF,20,8E,B0, 2657     <105>
1024 DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9
     ,14,A0,14,20,F2,B1,4C,A2,B1, 2665     <034>
1025 DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2
     ,20,3F,B2,90,9F,4C,8B,B5,A9, 2648     <123>
1026 DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8
     ,9D,00,D9,9D,00,DA,9D,00,DB, 2476     <237>
1027 DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2
     ,FF,20,D2,FF,98,4C,D2,FF,20, 2965     <160>
1028 DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00
     ,B1,5C,F0,06,20,D2,FF,C8,D0, 3100     <077>
1029 DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1
     ,FB,18,65,5A,85,5A,90,02,E6, 2606     <156>
1030 DATA 5B,06,5A,26,5B,C8,C0,08,90,EC,A5
     ,5A,65,5B,85,FF,60,18,A5,FB, 2467     <219>
1031 DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB
     ,C5,5F,A5,FC,E5,60,60,A0,B3, 3106     <183>
1032 DATA A9,FB,20,FF,B1,A0,01,B9,00,02,20
     ,D2,FF,CC,00,02,C8,90,F4,A9, 2692     <098>
1033 DATA 10,ED,00,02,AA,20,ED,B1,CA,D0,FA
     ,A5,62,20,4E,B1,A5,61,20,4E, 2453     <236>
1034 DATA B1,20,ED,B1,A5,60,20,4E,B1,A5,5F
     ,20,4E,B1,A9,9F,20,D2,FF,20, 2575     <038>
1035 DATA EA,B1,24,5E,10,01,60,A9,12,20,D2
     ,FF,A2,28,20,ED,B1,CA,D0,FA, 2646     <161>
1036 DATA A9,92,4C,D2,FF,A5,D6,C9,16,B0,01
     ,60,A9,A0,85,A4,A9,78,85,A6, 2945     <204>
1037 DATA A9,04,85,A5,85,A7,A2,13,A0,27,B1
     ,A4,91,A6,88,10,F9,CA,F0,19, 2671     <208>
1038 DATA 18,A5,A4,69,28,85,A4,90,02,E6,A5
     ,18,A5,A6,69,28,85,A6,90,E0, 2503     <251>
1039 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9
     ,0F,8D,18,D4,A9,00,8D,05,D4, 2776     <000>
1040 DATA A9,F7,8D,06,D4,A9,11,8D,04,D4,A9
     ,32,8D,01,D4,A9,00,8D,00,D4, 2413     <126>
1041 DATA A0,80,20,09,B3,A9,10,8D,04,D4,60
     ,A2,FF,CA,D0,FD,88,D0,F8,60, 2914     <240>
1042 DATA A9,0F,8D,18,D4,A9,2D,8D,05,D4,A9
     ,A5,8D,06,D4,A9,21,8D,04,D4, 2385     <119>
1043 DATA A9,07,8D,01,D4,A9,05,8D,00,D4,A0
     ,FF,20,09,B3,A9,20,8D,04,D4, 2250     <078>
1044 DATA A9,00,8D,01,D4,8D,00,D4,60,38,20
     ,F0,FF,8A,48,98,48,18,A0,06, 2179     <175>
1045 DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF
     ,B1,20,12,B3,20,E4,FF,F0,FB, 2931     <093>
1046 DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68
     ,A8,68,AA,18,4C,F0,FF,0D,0D, 2704     <088>
1047 DATA 0D,20,20,20,20,20,20,20,4D,41,53
     ,43,48,49,4E,45,4E,53,50,52, 1144     <216>
1048 DATA 41,43,48,45,20,2D,20,45,44,49,54
     ,4F,52,20,0D,0D,20,20,20,20, 1023     <038>
1049 DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D
     ,41,4E,4E,20,26,20,44,2E,57, 1128     <206>
1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20
     ,20,20,50,52,4F,47,52,41,4D, 1102     <117>
1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D
     ,20,20,20,53,54,41,52,54,41, 1073     <095>
1052 DATA 44,52,45,53,53,45,20,3A,20,24,00
     ,0D,0D,20,20,20,45,4E,44,41, 1014     <129>
1053 DATA 44,52,45,53,53,45,20,20,20,3A,20
     ,24,00,92,05,20,50,52,4F,47, 1171     <217>
1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20
     ,2A,2A,2A,20,46,41,4C,53,43, 1024     <027>
1055 DATA 48,45,20,45,49,4E,47,41,42,45,20
     ,2A,2A,2A,20,20,92,00,0D,0D, 1058     <098>
1056 DATA 2A,2A,2A,20,45,4E,44,45,20,2A,2A
     ,2A,00,13,05,20,20,12,44,92, 920      <148>
1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54
     ,92,41,50,45,0D,00,13,20,20, 1151     <035>
1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45
     ,52,00,20,D1,B1,20,48,B2,A0, 1606     <012>
1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC
     ,20,8E,B4,85,FB,C5,61,A5,FC, 3207     <251>
1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5
     ,60,B0,19,20,A7,B4,D0,14,60, 2860     <112>
1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0
     ,05,85,F8,4C,EF,B0,68,68,20, 2749     <088>
1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0
     ,09,20,D1,B1,20,48,B2,4C,0B, 2372     <046>
1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4
     ,20,EA,B1,20,0D,B5,24,5E,30, 2042     <120>
1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26
     ,20,9F,B2,24,5E,10,09,20,4E, 2435     <198>
1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20
     ,3F,B2,90,D7,A0,B4,A9,28,20, 2190     <207>
1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00
     ,85,5E,A5,61,85,FB,A5,62,85, 3056     <240>
1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E
     ,B1,A5,FB,85,FF,20,4E,B1,A9, 3003     <221>
1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1
     ,B1,FB,20,4E,B1,C8,C0,08,90, 2566     <070>
1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C
     ,A9,20,20,D2,FF,20,10,B2,A5, 2190     <059>
1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA
     ,B1,A9,FF,85,B8,85,B9,A9,04, 3073     <029>
1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20
     ,CC,FF,A9,FF,4C,C3,FF,20,5F, 3315     <189>
1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2
     ,A2,24,A9,2D,20,D2,FF,CA,D0, 2596     <111>
1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C
     ,C1,B4,20,B8,B5,A6,5F,A4,60, 2812     <015>
1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29
     ,BF,D0,03,4C,FB,B4,A9,01,20, 2577     <201>
1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF
     ,B1,20,F9,B1,4C,FB,B4,20,68, 2921     <237>
1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1
     ,A2,08,C9,44,F0,06,A2,01,C9, 2717     <213>
1077 DATA 54,D0,F1,A9,01,A8,20,BA,FF,A0,00
     ,E0,01,F0,1A,A9,40,8D,20,02, 2403     <101>
1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02
     ,C8,CC,00,02,90,F4,C8,C8,D0, 2182     <127>
1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02
     ,D0,F4,98,A2,20,A0,02,4C,BD, 2018     <025>
1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6
     ,B9,86,57,A9,01,20,C3,FF,A9, 2800     <022>
1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20
     ,B4,FF,A5,B9,20,96,FF,20,A5, 2911     <053>
1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5
     ,FF,85,62,20,AB,FF,A5,57,85, 2663     <214>
1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5
     ,86,5F,84,60,A5,BA,C9,01,D0, 2639     <131>
1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62
     ,4C,FB,B4,A9,13,20,D2,FF,A2, 2300     <120>
1085 DATA 1C,20,ED,B1,CA,D0,FA,60, 1230    <214>
```


MSE (Schluß). Dieses Listing können Sie (müssen aber nicht) mit dem Checksummer 64 V3 in dieser Ausgabe eingeben.

Ausführliches Directory

Um sich einen Überblick über den Inhalt einer Diskette zu verschaffen, benötigt man ein ausführliches Directory und einen Ausdruck der BAM. Mit diesem Listing können Sie beides haben.

Mit dem Programm »EX.DIR & BAM« können Sie übersichtlich und ausführlich Directory und BAM einer Diskette ausgeben. Die Ausgabe kann dabei wahlweise auf Bildschirm oder Drucker erfolgen. Sie müssen nur das Listing mit dem Checksummer abtippen.

Das Programm wird ganz normal mit RUN gestartet, da es sich um ein Basic-Programm handelt. Nach dem Starten kann man entscheiden, ob der Ausdruck auf Drucker oder Bildschirm erfolgen soll. Das Programm wurde für einen Star SG-10 mit Wiesemann-Interface geschrieben, es läuft aber auch mit allen Commodore-Druckern und Kompatiblen, da es über OPEN1,4,7 den Commodore-Groß/Klein-Modus ansteuert.

Nach der Wahl des Ausgabegerätes kommen Sie in ein Menü, in dem Sie entscheiden können, ob Sie die BAM oder das erweiterte Directory ausgegeben haben möchten. Die BAM wird sofort nach Eingabe ausgedruckt. Wollen Sie ein Directory haben, so werden Sie erst noch gefragt, ob die gelöschten Files mitangezeigt werden sollen oder nicht. Es werden aber nur die gelöschten Files des aktuellen Directoryblocks aufgelistet. Wenn man zum Beispiel die letzten acht Files einer Diskette gelöscht hat, so werden nicht mehr alle mitangezeigt, weil das Directory auf jeden Fall um einen Block kürzer geworden und somit dieser Block, in dem die Daten der gelöschten Files noch stehen, nicht mehr als Directoryblock gekennzeichnet ist.

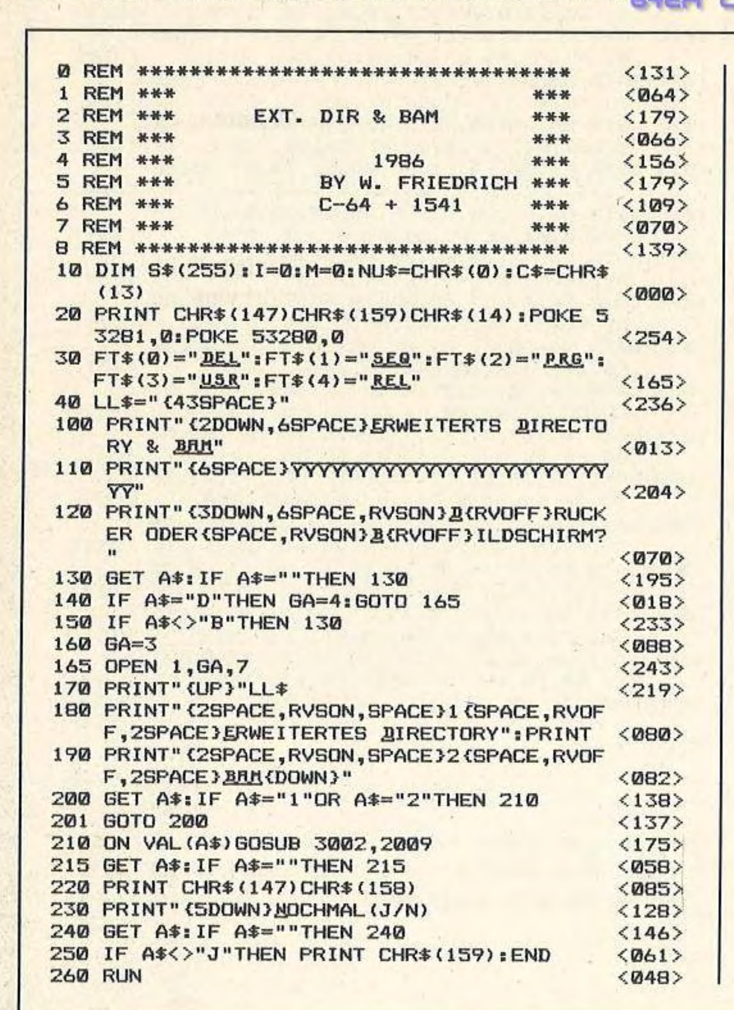
Hinweise zum Abtippen

Alle REM-Zeilen können weggelassen werden, da sie vom Programm nicht angesprungen werden. Allerdings leidet dadurch die Übersichtlichkeit ganz gewaltig. Der Leerstring 11$ in Zeile 40 sollte mindestens 29 Spaces enthalten, damit die Zeile mit der Abfrage des Ausgabegerätes vollständig gelöscht wird.

Wer das Programm für andere Drucker umschreiben will, kann sich auf die Zeile 165 beschränken, um die Sekundäradresse zu ändern. Allerdings sollte man die Filenummer 1 beibehalten, damit man nicht alle PRINT#-Statements ändern muß. Wem allerdings nicht gefällt, daß alle Filenamen klein geschrieben werden, muß größere Änderungen am Programm vornehmen. In Zeile 30 müssen alle Filetypen klein geschrieben werden. Zeile 165 muß folgendermaßen umgeschrieben werden: 165 OPEN1,GA. Man muß noch zwischen 205 und 210 die Zeile 207 PRINT CHR$(142) einfügen, und ab Zeile 2000 dürfen alle PRINT-Statements nur Kleinbuchstaben enthalten.

In Zeile 10 wurden die Variablen i und m mit Dummies belegt, damit sie ganz am Anfang der Variablenliste stehen und so schneller gefunden werden können.

(Wolfgang Friedrich/bs)



```
0 REM ********************************       <131>
1 REM ***                            ***     <064>
2 REM ***       EXT. DIR & BAM       ***     <179>
3 REM ***                            ***     <066>
4 REM ***            1986            ***     <156>
5 REM ***          BY W. FRIEDRICH   ***     <179>
6 REM ***          C-64 + 1541       ***     <109>
7 REM ***                            ***     <070>
8 REM ********************************       <139>
10 DIM S$(255):I=0:M=0:NU$=CHR$(0):C$=CHR$
   (13)                                      <000>
20 PRINT CHR$(147)CHR$(159)CHR$(14):POKE 5
   3281,0:POKE 53280,0                       <254>
30 FT$(0)="DEL":FT$(1)="SEQ":FT$(2)="PRG":
   FT$(3)="USR":FT$(4)="REL"                 <165>
40 LL$="{43SPACE}"                           <236>
100 PRINT"{2DOWN,6SPACE}ERWEITERTS DIRECTO
    RY & BAM"                                <013>
110 PRINT"{6SPACE}YYYYYYYYYYYYYYYYYYYYYYY
    YY"                                      <204>
120 PRINT"{3DOWN,6SPACE,RVSON}D{RVOFF}RUCK
    ER ODER{SPACE,RVSON}B{RVOFF}ILDSCHIRM?
    "                                        <070>
130 GET A$:IF A$=""THEN 130                  <195>
140 IF A$="D"THEN GA=4:GOTO 165              <018>
150 IF A$<>"B"THEN 130                       <233>
160 GA=3                                     <088>
165 OPEN 1,GA,7                              <243>
170 PRINT"{UP}"LL$                           <219>
180 PRINT"{2SPACE,RVSON,SPACE}1{SPACE,RVOF
    F,2SPACE}ERWEITERTES DIRECTORY":PRINT    <080>
190 PRINT"{2SPACE,RVSON,SPACE}2{SPACE,RVOF
    F,2SPACE}BAM{DOWN}"                      <082>
200 GET A$:IF A$="1"OR A$="2"THEN 210        <138>
201 GOTO 200                                 <137>
210 ON VAL(A$)GOSUB 3002,2009                <175>
215 GET A$:IF A$=""THEN 215                  <058>
220 PRINT CHR$(147)CHR$(158)                 <085>
230 PRINT"{5DOWN}NOCHMAL(J/N)                <128>
240 GET A$:IF A$=""THEN 240                  <146>
250 IF A$<>"J"THEN PRINT CHR$(159):END       <061>
260 RUN                                      <048>
```

```
500 REM UNTERPROGRAMME                       <188>
501 REM ***************                      <051>
505 REM DISKNAME UND ID AUSLESEN             <156>
506 REM *************************            <049>
510 MN=0                                     <216>
520 MN=MN+1:IF ASC(S$(143+MN))<>160 THEN D
    N$=DN$+S$(143+MN):GOTO 520               <027>
530 FOR I=0 TO 4:ID$=ID$+S$(162+I):NEXT      <193>
540 RETURN                                   <090>
600 REM FORMATIEREN                          <167>
601 REM ***********                          <208>
610 F$=STR$(INT(F))                          <232>
612 IF F=0 THEN 620                          <143>
615 IF F<1 THEN PRINT#1," ";                 <054>
620 FOR II=1 TO 5+SP-LEN(F$):PRINT#1," ";:
    NEXT                                     <012>
630 RETURN                                   <180>
700 REM STARTADDR PROG                       <244>
701 REM ***************                      <253>
710 OPEN 15,8,15,"I0":GOSUB 910              <172>
720 OPEN 2,8,2,"#":GOSUB 910                 <071>
730 PRINT#15,"U1:2,";0;FT;FS                 <020>
740 PRINT#15,"B-P:2,0"                       <234>
760 GET#2,A$,A$,L$:IF L$=""THEN L$=NU$       <077>
770 GET#2,H$:IF H$=""THEN H$=NU$             <117>
780 SD=256*ASC(H$)+ASC(L$):CLOSE 2:CLOSE 1
    5                                        <184>
790 RETURN                                   <086>
900 REM FEHLERKANAL                          <008>
901 REM ***********                          <254>
910 INPUT#15,Y1,Y$,Y2,Y3:IF Y1=0 THEN RETU
    RN                                       <203>
920 PRINT Y1;CHR$(18)Y$ CHR$(146),Y2,Y3:CL
    OSE 1:CLOSE 2:CLOSE 15:RETURN            <016>
1000 REM EINEN SEKTOR LESEN                  <224>
1001 REM ******************                  <026>
1010 OPEN 15,8,15,"I0":GOSUB 910             <218>
1020 OPEN 2,8,2,"#":GOSUB 910                <117>
1030 PRINT#15,"U1:2,""0";T;S                 <189>
1040 PRINT#15,"B-P:2,0"                      <026>
1050 PRINT#15,"M-R"CHR$(0)CHR$(5)            <149>
1060 GET#2,S$(0):IF S$(0)=""THEN S$(0)=NU$   <070>
```

```
1070 NT=ASC(S$(0))                                <214>
1080 GET#2,S$(1):IF S$(1)=""THEN S$(1)=NU$ <095>
1090 NS=ASC(S$(1))                                <230>
1110 FOR I=2 TO 255:GET#2,S$(I):IF S$(I)="
     "THEN S$(I)=NU$                              <037>
1120 NEXT                                         <114>
1130 CLOSE 2:CLOSE 15                             <034>
1999 RETURN                                       <023>
2000 REM BAM ANZEIGEN                             <194>
2001 REM ************                             <220>
2009 PRINT CHR$(147)                              <004>
2010 T=18:S=0:GOSUB 1010                          <002>
2012 GOSUB 510                                    <226>
2016 PRINT#1,"{7SPACE}"CHR$(18)CHR$(34)DN$
     CHR$(34)"{2SPACE}"ID$CHR$(146)               <156>
2018 PRINT#1,"{2SPACE}B"C$"{2SPACE}L"C$"{2
     SPACE}O"C$"{2SPACE}C"C$"{2SPACE}K 012
     345678901234567890 SEKTOR"C$                 <119>
2020 FOR I=1 TO 35                                <138>
2030 BF=ASC(S$(I*4))                              <157>
2040 FOR J=1 TO 3                                 <113>
2050 Q$=STR$(ASC(S$(I*4+J)))                      <091>
2060 GOSUB 2505                                   <137>
2070 B$=B$+BI$                                    <133>
2080 NEXT                                         <013>
2085 IF TZ<9 THEN PRINT#1," ";                    <133>
2090 TZ=TZ+1:PRINT#1,TZ;                          <012>
2100 IF TZ<18 THEN PRINT#1,LEFT$(B$,21)"{3
     SPACE}";:GOTO 2135                           <082>
2110 IF TZ<25 THEN PRINT#1,LEFT$(B$,19)"{5
     SPACE}";:GOTO 2135                           <056>
2120 IF TZ<31 THEN PRINT#1,LEFT$(B$,18)"{6
     SPACE}";:GOTO 2135                           <124>
2130 PRINT#1,LEFT$(B$,17)"{7SPACE}";              <071>
2135 IF BF<10 THEN PRINT#1," ";                   <022>
2140 PRINT#1,BF" BLOCKS FREE"                     <227>
2150 IF TZ<>18 THEN GF=GF+BF                      <002>
2160 B$="":NEXT                                   <064>
2170 PRINT#1,C$"{11SPACE}"GF" BLOCKS FREE" <220>
2180 RETURN                                       <206>
2500 REM DEZ->DUAL                                <148>
2501 REM *********                                <099>
2505 IF Q$=" 255"THEN BI$="........":RETUR
     N                                            <157>
2507 IF Q$=" 0"THEN BI$="********":RETURN <022>
2510 D=VAL(Q$):BI$="":Z=0                         <167>
2520 Z=Z+1:D=D/2:IF D<>INT(D)THEN BI$=BI$+
     ".":GOTO 2540                                <084>
2530 BI$=BI$+"*"                                  <135>
2540 IF D<>.5 THEN D=INT(D):GOTO 2520             <080>
2580 D=LEN(BI$):IF D=8 THEN RETURN                <006>
2590 FOR K=8-D TO 1 STEP-1                        <236>
2600 BI$=BI$+"*"                                  <207>
2610 NEXT                                         <080>
2620 RETURN                                       <138>
3000 REM EXT. DIR                                 <223>
3001 REM ********                                 <010>
3002 PRINT"{DOWN}DEL-FILES MIT ANZEIGEN?"; <187>
3003 GET A$:IF A$=""THEN 3003                     <197>
3004 IF A$="J"THEN DM=1:GOTO 3007                 <234>
3005 PRINT:DM=0                                   <187>
3007 PRINT CHR$(147)                              <242>
3010 T=18:S=0:GOSUB 1010                          <242>
3020 GOSUB 510                                    <218>
3030 PRINT#1,CHR$(18)CHR$(34)DN$CHR$(34)"{
     2SPACE}"ID$;CHR$(146)                        <195>
3040 IF NT<>18 OR NS>21 OR NS=0 THEN PRINT
     #1,664-GL" BLOCKS FREE":RETURN               <210>
3050 T=NT:S=NS:GOSUB 1010                         <201>
3060 FOR I=0 TO 7                                 <079>
3070 M=I*32+2:MN=0:NF$=""                         <020>
3080 FT=ASC(S$(M+1)):FS=ASC(S$(M+2)):TF=AS
     C(S$(M))AND 15:SD=0                          <042>
3085 IF DM=0 AND FT=0 THEN 3200                   <111>
3090 MN=MN+1:TE=ASC(S$(M+2+MN)):IF TE<>160
      AND TE>10 THEN NF$=NF$+S$(M+2+MN):GO
     TO 3090                                      <178>
3095 IF NF$=""THEN 3200                           <075>
3100 L=ASC(S$(M+29))*256+ASC(S$(M+28)):LG=
     LG+L                                         <180>
3120 IF TF<>0 THEN GOSUB 710                      <160>
3130 F=L:SP=0:GOSUB 610                           <002>
3150 PRINT#1,L"{3SPACE}"CHR$(34)NF$;CHR$(3
     4);LEFT$(LL$,18-LEN(NF$));                   <253>
3160 PRINT#1,FT$(TF);"{2SPACE}";                  <090>
3165 IF FT<10 THEN PRINT#1," ";                   <037>
3170 PRINT#1," ";FT;                              <106>
3175 IF FS<10 THEN PRINT#1," ";                   <031>
3180 PRINT#1," ";FS;"{3SPACE}"LEFT$(LL$,6-
     LEN(STR$(SD)));SD;                           <145>
3185 F=L/4:SP=2:GOSUB 610:PRINT#1,L/4             <000>
3190 IF TF<>0 THEN GL=GL+L                        <132>
3200 NEXT                                         <162>
3210 GOTO 3040                                    <246>
```

Listing »EX.DIR & BAM«. Beachten Sie bitte die Eingabehinweise auf Seite 97.

Autostart C 128

Dieses Programm veranlaßt den C128 beim Einschalten oder einem Reset dazu, ein Basic-Programm zu laden und zu starten.

Das Pinzip des Autostarts beruht auf dem Vorhandensein eines Boot-Sektors, der vom C 128 bei jedem Kaltstart (Reset oder Einschalten) auf der Diskette gesucht wird. Dabei handelt es sich um den Sektor 0 auf Spur 1, den das Programm manipuliert.

Nach dem Start des Generatorprogramms (Listing) erscheint ein Titelbild, auf dem eine kurze Erklärung ausgegeben wird.

Drückt man nun eine Taste, so wird auf die Diskette zugegriffen, um festzustellen, ob der Boot-Sektor bereits belegt ist. Dies geschieht mit Hilfe des Block-Allocate-Befehls des DOS, der normalerweise einen Block als belegt kennzeichnet. Wird dieser Befehl auf einen schon belegten Block angewendet, so meldet das DOS einen NO BLOCK-Fehler. Ist dies der Fall, so stellt das Programm drei Möglichkeiten zur Auswahl:

1. Block freigeben

Dies sollte man nur wählen, wenn man sich ganz sicher ist, daß dieser Block nicht belegt ist. Andernfalls kann ein auf der Diskette befindliches Programm zerstört werden.

2. Erneuter Versuch

Bei Lesefehlern können hiermit weitere Leseversuche ausgeführt werden.

3. Abbrechen

Es kann nun eine Nachricht eingegeben werden, die der Computer beim Laden ausgibt. Diese kann in Anführungszeichen stehen und damit auch Steuerzeichen enthalten.

Anschließend werden Sie aufgefordert, den Namen des zu ladenden Programms einzugeben.

Ist dies geschehen, schreibt der Computer den Block in den Puffer #6 und veranlaßt anschließend die Floppy-Station, diesen auf Diskette zu speichern.

Der Boot-Sektor besitzt die folgende Aufteilung:

Byte-Nummer:	Inhalt:	Bemerkung:
0,1,2	CBM	daran erkennt das Betriebssystem den Boot-Sektor
3,4	0,0	Speicheradresse der Folgeboot-Sektoren. Diese beiden Werte werden vom Programm nicht genutzt, da nur ein Sektor gebildet wird. Hier wäre auch denkbar, mehrere Boot-Sektoren zu laden.
5	0	Konfigurationsindex, der beim Laden von folgenden Boot-Sek-

		toren in die MMU gespeichert werden soll.
6	0	Hier steht die Anzahl der Boot-Sektoren, die noch zu laden sind.
7 bis (0)	Text	Die nun folgende Byte-Folge wird über die PRIMM-Routine ausgegeben (Abschluß des Textes mit CHR$(0)).
PRG-Name		Es schließt sich der Programmname an, der ebenfalls von einem Null-Byte abgeschlossen wird.
Maschinen-PRG		Nach der Testausgabe und dem Laden wird die folgende Maschinenroutine angesprungen.

Nach Schreiben des Boot-Sektors wird das zu ladende Programm auf seine Anfangsadresse hin untersucht. Ist diese nicht unter $1C01 (normaler Basic-Start), so erfolgt die Ausgabe einer Meldung (dies ist zum Beispiel oft bei Compilaten des »BASIC 128« der Fall). In so einem Fall ist es nötig, das zu bootende Programm nach Abschluß von »Autostart 128« mit DLOAD zu laden, mit SCRATCH zu löschen und über DSAVE wieder unter dem gleichen Namen auf Diskette zu speichern.

Der Grund ist darin zu suchen, daß das Betriebssystem das zu ladende File absolut lädt. Man könnte also auch Maschinenspracheroutinen oder Interfaces bei jedem Reset neu von Diskette in den User-Bereich laden.

Ist dieser Test beendet, so schließt das Programm alle Files und startet sich neu.

Nachdem ein Boot-Sektor auf Diskette geschrieben wurde, sollte kein COLLECT beziehungsweise VALIDATE mehr durchgeführt werden, da der Boot-Sektor wieder als nicht belegt gekennzeichnet und überschrieben werden würde. Sollte dies doch unumgänglich sein, so muß nach COLLECT »Autostart 128« neu angewendet oder der folgende Befehl eingegeben werden:

```
OPEN 1,8,15,"B-A:0 1 0":CLOSE 1
```

(Dirk Paessler/dm)

```
1000 COLOR 0,16: COLOR 4,16: COLOR 5,12
1010 TA$="{10SPACE}": TA$=TA$+TA$: TA$=TA$+T
     A$
1020 GN=8                              : RE
     M " GERFTENUMMER
1030 GOSUB 1440
1040 RESTORE 1530
1050 PRINT CHR$(14) CHR$(7) CHR$(8)
1060 PRINT "{2DOWN,2SPACE}DIESES PROGRAMM GE
     NERIERT EINEN BOOT-
1070 PRINT "{2SPACE}SECTOR AUF TRACK {RVSON}
     1{RVOFF} SECTOR {RVSON}0{RVOFF} , DER D
     EN
1080 PRINT "{2SPACE}C= 128 BEIM EINSCHALTEN
     VERANLAVT EIN
1090 PRINT "{2SPACE}BESTIMMTES BASIC - PROGR
     AMM{2SPACE}ZU LADEN": PRINT "{2SPACE}UN
     D ZU STARTEN.{3DOWN}
1100 A$="(C)OPYRIGHT{2SPACE}BY": GOSUB 1470:
      A$="DIRK PAESSLER": GOSUB 1470
1110 A$="LFRCHENWEG{2SPACE}8": GOSUB 1470
1120 A$="8520 ERLANGEN": GOSUB 1470: PRINT :
      PRINT : PRINT : PRINT
1130 GOSUB 1500
1140 GOSUB 1440
1150 OPEN 15,GN,15,"I"                  : R
     EM " BEFEHLSKANAL ERCFFNEN UND DISK-IN.
1160 OPEN 1,GN, 6,"#"                   : RE
     M " DIREKTKANAL ERCFFNEN
1170 PRINT#15,"B-A:0 1 0": IF DS<>65 THEN 12
     40: REM " IST DER BLOCK NOCH FREI ???
1180 A$="BLOCK 1:0 IST BELEGT": GOSUB 1470
1190 PRINT "{CTRL+G,DOWN} <F> ... BLOCK FREI
     GEBEN
1200 PRINT "{DOWN} <N> ... NOCHEINMAL VERSUC
     HEN
1210 PRINT "{DOWN} <A> ... VORGANG ABBRECHEN
     {2DOWN}
1220 GOSUB 1500: IF A$="A" THEN 1400: ELSE :
      IF A$="N" THEN 1170
1230 IF A$<>"F" THEN PRINT "{CTRL+G,UP}";: G
     OTO 1220
1240 PRINT#15,"B-F:0 1 0"               : RE
     M " BLOCK WIEDER FREIGEBEN
1250 PRINT "{2DOWN,CTRL+G,RVSON}BOOTNACHRICH
     T :{DOWN}": INPUT A1$
1260 PRINT "{2DOWN,CTRL+G,RVSON}PROGRAMMNAME
      :{DOWN}": INPUT PR$
1270 PRINT#15,"B-P 6 0": PRINT#1,"CBM" CHR$(
     0) CHR$(0) CHR$(0) CHR$(0);
1280 PRINT#1,A1$;CHR$(0);PR$;CHR$(0);
1290 READ A$: DO WHILE A$<>"ENDE": PRINT#1,C
     HR$(DEC(A$));: READ A$: LOOP
1300 PRINT#15,"U2 6 0 1 0"
1310 IF DS=0 THEN 1330
1320 PRINT "{CTRL+G,RVSON}SCHREIBFEHLER{RVOF
     F} : "DS$: GOSUB 1500: GOTO 1400
1330 PRINT#15,"B-A:0 1 0"               : RE
     M " BLOCK BELEGEN
1340 CLOSE 1: OPEN 1,8,6,PR$+",P,R": IF DS=0
      THEN 1360
1350 PRINT "{CTRL+G,RVSON}DISKFEHLER{RVOFF}
     : "DS$: GOSUB 1500: GOTO 1400
1360 GET #1,A$,B$: A=ASC(A$+CHR$(0))+ASC(B$+
     CHR$(0))*256
1370 IF A=DEC("1C01") THEN 1390
1380 PRINT "STARTADRESSE DES PROGRAMMS NICHT
      RICHTIG": PRINT : PRINT : GOSUB 1500
1390 PRINT : PRINT : A$="AUFTRAG VOLLENDET":
      GOSUB 1470: PRINT : PRINT : PRINT : GO
     SUB 1500
1400 REM "PRG ENDE
1410 CLOSE 1: CLOSE 15: GOTO 1030
1420 END
1430 :
1440 REM "KOPFZEILE
1450 PRINT "{2HOME,CLR,RVSON}"TA$"{HOME}";:
     A$="AUTOSTART 128": GOSUB 1470: PRINT :
      PRINT : RETURN
1460 :
1470 REM "ZEILE AUF 40 ZEICHEN ZENTRIEREN
1480 A$=LEFT$(TA$,(40-LEN(A$))/2)+A$: PRINT
     A$
1490 RETURN
1500 REM " ZUM TASTENDRUCK AUFFORDERN
1510 A$="BITTE EINE TASTE DRYCKEN": GOSUB 14
     70: GET KEY A$: RETURN
1520 :
1530 :
1540 REM "MASCHINENPROGRAMM SPEICHERT RUN <C
     R> IN DER TASTATURPUFFER
1550 DATA A2,04      : REM "LDX #$04{7SPACE}AN
     ZAHL DER ZEICHEN
1560 DATA BD,2D,10 : REM "LDA $102D,X←{2SPA
     CE}AUS FUNKTIONSTASTENTABELLE LESEN
1570 DATA 9D,49,03 : REM "STA $0349,X {2SPA
     CE}UND IN TASTATURPUFFER SPEICHERN
1580 DATA CA         : REM "DEX{9SPACE}
1590 DATA D0,F7      : REM "BNE *********{2SPA
     CE}SCHLEIFE ZU ENDE
1600 DATA A9,04      : REM "LDA #$04{7SPACE}AN
     ZAHL DER ZEICHEN
1610 DATA 85,D0      : REM "STA $D0{8SPACE}IN
     ZAHL DER ZEICHEN IM PUFFER
1620 DATA 60         : REM "RTS{12SPACE}UND EN
     DE
1630 DATA ENDE
```

Listing »Autostart 128«. Bitte im 128er-Modus eingeben.

Diskmonitor C128

Der Monitor des C128 ist schon eine praktische Angelegenheit. Leider fehlte bisher der Diskmonitor. Hier ist er nun.

Eines der wichtigsten Werkzeuge im Umgang mit Disketten ist der Diskmonitor, der es gestattet, jede Spur und jeden Sektor anzusehen, zu verändern und wieder auf Diskette zurückzuschreiben. Der im C128 eingebaute Monitor hat schon einige für diese Zwecke benötigte Befehle:

1. Das Senden von Befehlen an die Diskettenstation
2. Laden des Disketten-Inhaltsverzeichnisses
3. Einen Lineassembler
4. Einen Disassembler und
5. Das Anzeigen von Speicherbereichen.

Das Wichtigste aber, Befehle zum Einlesen und Zurückschreiben einzelner Blöcke, ist leider nicht vorhanden.

Die Aufgabenstellung ist also, diese beiden Befehle in den vorhandenen Monitor mit einzubinden.

Da Commodore seit Jahren bei ihren Betriebssystemen mit Sprunglisten und Vektoren arbeitet, bestand die berechtigte Hoffnung, daß beim Monitor ebenfalls ein Vektor vorhanden ist. Nachdem der Autor schon eine ganze Weile im ROM auf der Suche nach den Vektoren mit Erfolg gestöbert hatte, erschien dann bei Markt&Technik eine wahre Fundgrube – das Buch »C128-ROM-Listing: Operating System« von Schineis, Braun und Demgensky, das die Sache wesentlich erleichterte. Das Buch ist allen zu empfehlen, da es für alle Sprungbefehle Label angibt, die am Ende der gut kommentierten Listings alphabetisch sortiert mit ihren Adressen aufgeführt sind.

Da existiert nun solch ein Vektor mit dem Namen IMONCD ($032E), der auf die Monitorroutine »Kommando in Tabelle suchen und dann ausführen« zeigt, also der ideale Ansatzpunkt ist, um eigene Routinen einzubinden.

Als Befehlszeichen wurde der Pfeil nach oben »↑« mit einem zusätzlichen Buchstaben als Befehlszeichen gewählt.

»↑«: Zeige eingelesenen Block an (der Block steht im Bereich $1300 bis $13FF in BANK 0).

»↑R« (Spur Sektor Gerät Laufwerk): Lies einen Block von Diskette und zeige ihn an.

»↑W« (Spur Sektor Gerät Laufwerk): Schreibe den Block auf Diskette.

»↑N«: Lies nächsten Block von Diskette und zeige ihn an.

»↑G« (Gerät Laufwerk): Spezifiziere neue Geräteadresse (eingestellt ist Gerät 8).

»↑D« (Laufwerk): Spezifiziere neue Laufwerksnummer (Eingestellt ist Laufwerk 0 – dieser Befehl wird für Doppellaufwerke benötigt).

»↑L«: Lese den vorherigen Block wieder von Diskette ein.

»↑+«: Lese den physikalisch nächsten Block von der Diskette.

»↑-«: Lese den physikalisch vorhergegangenen Block von der Diskette.

»↑P« (DEV): Leitet die Ausgabe vom Bildschirm auf Gerät DEV (vorzugsweise auf den Drucker) um.

»↑B«: Ausgabe wieder auf Bildschirm.

Nach jedem Zugriff auf die Floppy wird der Fehlerkanal ausgelesen und in der Zeile, die dem Befehl folgt, ausgegeben.

Eingabehinweise

Bitte geben Sie das Programm (Listing) mit dem MSE im C64-Modus ein und speichern es. Das Programm wird in Zukunft mit dem Befehl

`SYS DEC("1400")`

initialisiert. Wird dann bei Bedarf der Monitor mit dem Befehl MONITOR aufgerufen, stehen auch die DISKMON-Befehle zur Verfügung.

(Michael Bauer/dm)

```
Name : diskmon.exe            1400 1769
---------------------------------------
1400 : ad 2e 03 8d 3b 14 ad 2f  a0
1408 : 03 8d 3c 14 a9 43 8d 2e  ab
1410 : 03 a9 14 8d 2f 03 20 7d  25
1418 : ff 0d 44 49 53 4b 4d 4f  3b
1420 : 4e 20 49 4e 49 54 49 41  79
1428 : 4c 49 53 49 45 52 54 07  5d
1430 : 00 a2 ff 4c 3f 4d 20 cc  83
1438 : ff 68 4c 06 b0 4c 8b b0  3c
1440 : 4c bc b0 48 c9 40 f0 ee  60
1448 : c9 4c f0 ea c9 53 f0 e6  99
1450 : c9 56 f0 e2 20 f3 16 68  a8
1458 : c9 5e d0 de 20 e9 b8 f0  76
1460 : 27 a2 0a dd 25 17 d0 0c  7d
1468 : 8e b2 0a bd 30 17 48 bd  e2
1470 : 3b 17 48 60 ca 10 ec 30  96
1478 : c7 ad 65 17 8d 63 17 ad  fe
1480 : 64 17 8d 62 17 4c c4 14  2f
1488 : 20 7d ff 0d 53 50 55 52  ba
1490 : 3a 20 00 ad 62 17 20 a5  3b
1498 : b8 20 7d ff 20 20 53 45  9a
14a0 : 4b 54 4f 52 3a 20 00 ad  33
14a8 : 63 17 20 a5 b8 a2 0c bd  a0
14b0 : 54 17 9d 00 02 ca 10 f7  9e
14b8 : e8 86 7a 20 e9 b8 4c 3a  90
14c0 : 14 20 5f 16 20 cc ff a9  3b
14c8 : 00 aa 86 62 20 68 ff a0  92
14d0 : 0f ae 60 17 20 ba ff a9  5c
14d8 : 00 20 bd ff 20 c0 ff b0  c1
14e0 : 35 a9 0d a8 ae 60 17 20  cd
14e8 : ba ff a9 01 a2 53 a0 17  a2
14f0 : 20 bd ff 20 c0 ff b0 1e  fe
14f8 : ad b2 0a f0 64 a9 31 20  38
1500 : b5 16 a2 0d 20 c6 ff a2  88
1508 : 00 20 cf ff 9d 00 13 a5  7e
1510 : 90 d0 03 e8 d0 f3 20 cc  ad
1518 : ff a9 0d 38 20 c3 ff 20  97
1520 : b4 b8 a2 00 20 c6 ff b0  73
1528 : 18 20 cf ff 8d 68 17 d0  5e
1530 : 03 20 cf ff c9 0d f0 09  12
1538 : 20 d2 ff a5 90 29 bf f0  a9
1540 : f0 20 cc ff 20 f3 16 a9  c1
1548 : 00 38 20 c3 ff ad 68 17  22
1550 : c9 30 d0 0a ad b2 0a c9  d3
1558 : 06 90 03 4c 88 14 4c 3d  c6
1560 : 14 a2 0d 20 c9 ff b0 ae  c9
1568 : a2 01 bd 00 13 20 d2 ff  78
1570 : b0 a4 e8 d0 f5 ad 00 13  b9
1578 : 20 d2 ff b0 99 20 cc ff  e5
1580 : a9 32 20 b5 16 4c 16 15  47
1588 : ad 00 13 f0 18 ae 62 17  c7
1590 : 8e 64 17 ae 63 17 8e 65  e0
1598 : 17 8d 62 17 ad 01 13 8d  3c
15a0 : 63 17 4c c4 14 20 7d ff  73
15a8 : 0d 4b 45 49 4e 20 42 4c  5d
15b0 : 4f 43 4b 20 4d 45 48 52  3c
15b8 : 07 00 4c 3d 14 20 8f 16  27
15c0 : 4c 3d 14 20 a1 16 4c 3d  2a
15c8 : 14 20 0e 16 20 b4 b8 20  fe
15d0 : a7 b7 d0 03 a2 04 2c a6  30
15d8 : 60 e0 04 b0 03 4c 40 14  7b
15e0 : e0 08 b0 f9 8e 66 17 a9  fb
15e8 : 04 a0 00 20 ba ff a9 00  93
15f0 : 20 bd ff 20 c0 ff 90 03  47
15f8 : 4c 3d 14 a2 04 20 c9 ff  a5
1600 : a9 01 8d 67 17 4c 3d 14  6b
1608 : 20 0e 16 4c 3d 14 ad 67  38
1610 : 17 f0 0e 20 cc ff a9 04  a2
1618 : 38 20 c3 ff a9 00 8d 67  f1
1620 : 17 60 ae 63 17 e8 8e 63  39
1628 : 17 20 04 17 cd 63 17 b0  e9
1630 : 0c ae 62 17 e8 8e 62 17  ca
1638 : a2 00 8e 63 17 4c c4 14  f9
1640 : ae 63 17 f0 07 ca 8e 63  4b
1648 : 17 4c c4 14 ae 62 17 ca  29
1650 : d0 01 e8 8e 62 17 20 04  14
1658 : 17 8d 63 17 4c c4 14 20  6d
1660 : a7 b7 f0 4d a5 60 f0 4a  7e
1668 : c9 47 b0 46 ae 62 17 8e  41
1670 : 64 17 8d 62 17 20 a7 b7  90
1678 : f0 37 ae 63 17 a5 60 8d  57
1680 : 63 17 20 04 17 cd 63 17  93
1688 : f0 02 90 26 8e 65 17 20  13
1690 : a7 b7 f0 1d a6 60 e0 08  f4
1698 : 90 18 e0 20 b0 14 8e 60  17
16a0 : 17 20 a7 b7 f0 0b a5 60  67
16a8 : f0 04 c9 01 d0 04 8d 61  53
16b0 : 17 60 4c 40 14 8d 47 17  0b
16b8 : ad 62 17 20 fb f9 8e 4e  c7
16c0 : 17 8d 4f 17 ad 63 17 20  e7
16c8 : fb f9 8e 51 17 8d 52 17  e3
16d0 : ad 61 17 20 fb f9 8d 4c  56
16d8 : 17 a2 00 20 c9 ff 90 03  29
16e0 : 4c 16 15 a2 00 bd 46 17  06
16e8 : 20 d2 ff e8 e0 0d d0 f5  34
16f0 : 4c cc ff ad 67 17 f0 0b  61
16f8 : 20 cc ff 20 b4 b8 a2 04  26
1700 : 20 c9 ff 60 ad 62 17 a0  9c
1708 : 07 d9 15 17 90 03 88 10  87
1710 : f8 b9 1d 17 60 47 42 3c  d1
1718 : 35 24 1f 19 12 10 11 12  54
1720 : 14 10 11 12 14 57 4d 47  83
1728 : 44 50 42 52 4e 4c 2b 2d  be
1730 : 14 14 15 15 15 16 14 15  b3
1738 : 14 16 16 c0 87 bc c2 c8  f0
1740 : 07 c0 87 78 21 3f 55 31  5c
1748 : 3a 31 33 20 30 20 31 38  25
1750 : 20 30 30 23 4d 20 31 33  fa
1758 : 30 30 20 31 33 46 46 00  4d
1760 : 08 00 12 01 12 00 04 00  3e
1768 : 30 6e 37 13 ae 28 13 bd  f4
```

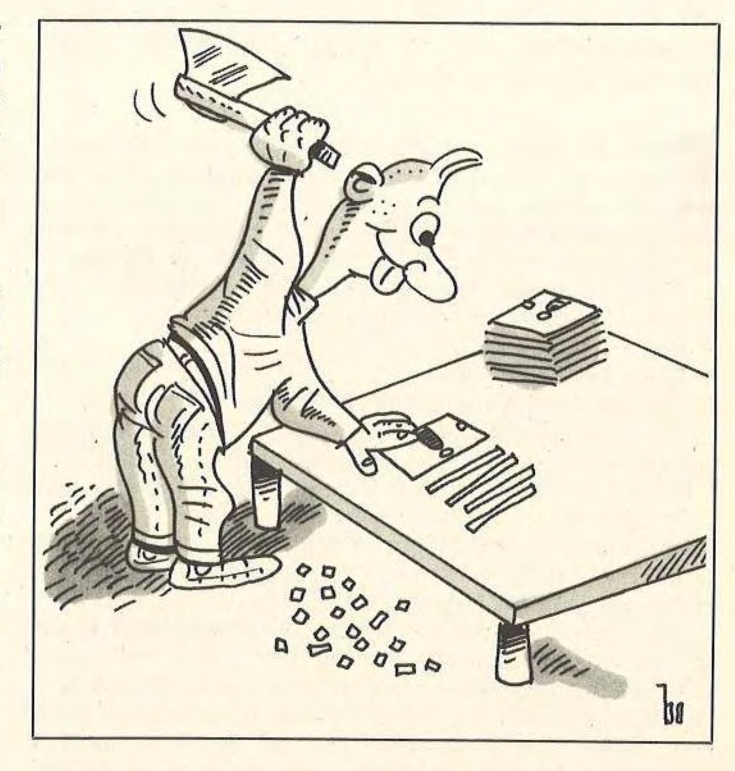
Listing »Diskmon«. Es erweitert den Monitor im C128 zum Diskmonitor. Bitte mit dem MSE (Seite 100) im C64-Modus eingeben.

Radikal gelöscht

Datenschutz ist eines der meist benutzten Wörter unserer Zeit. In politischen Diskussionen oder im Zusammenhang mit dem Thema »Raubkopien« fällt dieser Begriff immer wieder. Mit dem »Physical Scratcher« können auch Sie sich vor Datenklau schützen!

Es soll schon passiert sein, daß Spione die Papierkörbe von Regierungsbeamten durchsucht haben und dabei auf wichtige Informationen stießen. Das könnte Ihnen auch passieren: Sie löschen ein wichtiges Programm auf einer Diskette mit dem Scratch-Befehl und geben diese weiter – und jemand anders rekonstruiert das Programm wieder. Der Scratch-Befehl ändert nämlich nur den Directory-Eintrag, löscht das Programm aber nicht wirklich. Ein findiger Kopf kann jederzeit, nur mit einem Disketten-Monitor bewaffnet, das Programm wieder hervorzaubern.

»Physical Scratcher« hingegen löscht die Programme vollständig und unwiederbringbar. Ein mit »Physical Scratcher« durchgeführter Löschvorgang ist nicht mehr rückgängig zu machen! Seien Sie also extrem vorsichtig bei der Anwendung des Listings.



Daten ausradiert

Alles, was Sie eintippen müssen, ist das relativ kurze Listing. Bitte verwenden Sie dazu den MSE.

Die Anwendung des Programmes ist sehr einfach. Nach dem Start mit »RUN« müssen Sie nur den Namen des zu löschenden Programms angeben. Er darf maximal 18 Zeichen lang sein, die letzten zwei Zeichen bestimmen den Filetyp (»,p«,»,s«).

Danach folgt eine Sicherheitsabfrage mit den Eingabe-Möglichkeiten <Y>, <N> und <Q>. <Q> bedeutet Quit und steht für das Verlassen des Programms. Bei Eingabe von <Y> wird gelöscht, bei Eingabe von <N> (wenn Sie vorher einen Tippfehler gemacht haben) hingegen nicht.

Das Löschen ist ziemlich schnell, da ein Programm in der Floppystation diese Aufgabe übernimmt. Dabei werden die vom zu löschenden Programm belegten Blöcke mit 00-Byte überschrieben. Nach dem Löschvorgang wird zum Anfang des Programms gesprungen.

Die Idee zu diesem Programm stammt übrigens von einer ähnlichen Routine des IBM-PCs.

Nochmals eine Warnung: Mit »Physical Scratcher« gelöschte Programme sind unwiederbringlich verloren, seien Sie also vorsichtig bei der Anwendung! (bs)

```
Name : physical scratch  0801 0a45
-----------------------------------
0801 : 0c 08 ff ff 9e 20 32 31  27
0809 : 30 30 00 00 00 45 41 53  27
0811 : 53 20 42 59 20 49 48 20  de
0819 : 28 43 29 20 38 35 00 2e  bb
0821 : 08 46 00 2e 4f 50 54 20  1b
0829 : 50 2c 4f 32 00 34 08 50  0c
0831 : 00 3b 00 a2 00 8e 20 d0  ba
0839 : 8e 21 d0 86 02 86 c6 a9  1f
0841 : 53 a2 3a 8d 00 02 8e 01  72
0849 : 02 a9 cf a0 09 20 1e ab  89
0851 : a2 00 86 cc ca 86 d4 a5  ae
0859 : c6 f0 f5 20 b4 e5 c9 14  e2
0861 : d0 0c a4 02 f0 ea c6 02  26
0869 : 20 d2 ff 4c 51 08 c9 0d  12
0871 : f0 11 a6 02 e0 12 f0 d8  e8
0879 : e6 02 9d 02 02 20 d2 ff  74
0881 : 4c 51 08 e6 cc a9 00 85  7a
0889 : cf a9 20 ae 87 02 20 13  3a
0891 : ea a9 0d a0 0a 20 1e ab  19
0899 : a5 c5 c9 40 f0 fa c9 19  db
08a1 : f0 18 c9 27 d0 03 4c 34  b3
08a9 : 08 c9 3e d0 eb a9 00 85  57
08b1 : c6 a2 58 9d 00 02 ca d0  6f
08b9 : fa 60 a9 02 8d 20 d0 a5  f6
08c1 : 02 a2 02 a0 02 20 bd ff  c1
08c9 : a9 02 a2 08 a0 60 20 ba  20
08d1 : ff 20 c0 ff a9 08 20 b4  d5
08d9 : ff a9 60 20 96 ff 20 a5  fe
08e1 : ff a5 90 4a 4a 90 1a a9  05
08e9 : 2f a0 0a 20 1e ab a5 c5  50
08f1 : c9 01 d0 fa a9 08 20 ab  81
08f9 : ff a9 02 20 c3 ff 4c 34  27
0901 : 08 a9 08 20 ab ff a9 01  47
0909 : a2 08 a0 6f 20 ba ff a9  f0
0911 : 00 20 bd ff 20 c0 ff a2  de
0919 : 00 a9 08 20 b1 ff a9 6f  95
0921 : 20 93 ff a0 02 b9 27 0a  bd
0929 : 20 a8 ff 88 10 f7 8a 20  d9
0931 : a8 ff a9 05 20 a8 ff a9  7e
0939 : 01 20 a8 ff bd 9a 09 20  89
0941 : a8 ff a9 08 20 ae ff e8  9d
0949 : e0 38 d0 cd a9 08 20 b1  f2
0951 : ff a9 6f 20 93 ff a2 00  c8
0959 : bd 2a 0a 20 a8 ff e8 e0  a2
0961 : 06 d0 f5 a9 08 20 ae ff  be
0969 : a9 02 20 c3 ff a9 01 20  25
0971 : c3 ff a6 02 e8 e8 8a a2  63
0979 : 00 a0 02 20 bd ff a9 01  d2
0981 : a2 08 a0 6f 20 ba ff 20  55
0989 : c0 ff a9 01 20 c3 ff a9  47
0991 : 00 8d 20 d0 85 c6 4c 4a  ce
0999 : 08 ad 00 04 48 ad 01 04  f6
09a1 : 48 a5 18 85 08 a5 19 85  90
09a9 : 09 a2 02 a9 00 9d 00 04  ae
09b1 : e8 d0 fa a9 90 85 01 a5  7a
09b9 : 01 30 fc 68 85 09 68 f0  43
09c1 : 0c 85 08 a9 80 85 01 a5  4b
09c9 : 01 30 fc 10 cc 60 93 1e  7e
09d1 : 11 12 20 50 48 59 53 49  2d
09d9 : 43 41 4c 20 53 43 52 41  ef
09e1 : 54 43 48 45 52 20 56 31  73
09e9 : 2e 30 20 42 59 20 49 2e  98
09f1 : 48 45 4e 44 52 49 43 4b  0b
09f9 : 53 20 0d 45 4e 54 45 52  8a
0a01 : 20 46 49 4c 45 20 4e 41  31
0a09 : 4d 45 3a 00 0d 11 99 41  ca
0a11 : 52 45 20 59 4f 55 20 53  00
0a19 : 55 52 45 3f 20 28 59 2f  d8
0a21 : 4e 2f 51 29 1e 00 57 2d  1a
0a29 : 4d 4d 2d 45 00 05 0d 11  8f
0a31 : 46 49 4c 45 20 44 4f 45  c4
0a39 : 53 20 4e 4f 54 20 45 58  26
0a41 : 49 53 54 00 90 02 e6 1d  38
```

Listing zu Physical Scratcher. Bitte mit dem MSE (auf Seite 100) eingeben.

Directory unter Druck

Übersicht in der Programmsammlung läßt sich mit »Catalog Printer+« schaffen, dem idealen Programm zum Ausdrucken umfangreicher Directories.

Wer sich schon einmal mit »LOAD"$",8« durch Hunderte von Disketten gesucht hat, nur um ein einzelnes Programm zu finden, wird erleichtert das folgende Listing abtippen. »Catalog Printer+« druckt ganz sauber und übersichtlich Ihre Directories auf einem Epson-Drucker. Einen Beispielausdruck in Originalgröße zeigt Bild 1.

Ein ähnliches Programm konnten Sie schon im Sonderheft 2/86 finden, doch das dortige Listing wurde für dieses Sonderheft von einem Leser total überarbeitet und sehr stark verbessert. Die neue Version arbeitet wesentlich schneller und berücksichtigt auch schreibgeschützte Files.

Nach dem Programm-Start wird in Zeile 30 und 40 überprüft, ob ein Drucker angeschlossen ist. Das Programm ist auf Epson-Drucker der Serien RX und FX sowie Kompatible ausgelegt. Für andere Drucker müssen Sie wahrscheinlich Änderungen in den Zeilen 610 und 620 vornehmen. Damit das Programm mit Druckern funktioniert, die keine Kleinschrift auf dem Drucker ausgeben können (MPS 801, 802, 803, und ähnliche), wären umfangreiche Änderungen im gesamten Listing notwendig. Dann wäre auch nur ein ein- oder zweispaltiger Druck möglich. In der abgedruckten Version ist das Programm nicht mit diesen Druckern lauffähig.

Nach der Überprüfung auf angeschlossenen Drucker meldet sich dann ein Menü, von dem aus Sie mit den Funktionstasten die einzelnen Programmteile starten können. Um die Directories mit der richtigen Anzahl belegter Blöcke auszugeben, ist es möglich, vor dem Ausdruck zu validieren, das heißt die Diskette »aufzuräumen«. (Genaue Informationen zum Validieren finden Sie im Floppy-Kurs in diesem Sonderheft.)

Mit Druck auf <RETURN> oder <F1> wird erst einmal nach dem Tagesdatum gefragt und dann der Ausdruck des Directorys begonnen. Je nach Anzahl der Einträge kann es ein- bis dreispaltig werden. Das ausgedruckte Directory läßt sich nun beispielsweise ausschneiden und auf die Papierhülle der Diskette kleben. Damit hat man schnelle Übersicht über den kompletten Disketteninhalt, ohne ein Directory umständlich laden zu müssen. (Herbert Kaufhold/bs)

NAME : AUSGABE 7/86   ID: 64   17 BLOECKE FREI.   STAND VOM: 14.07.86

23	LESER-INFO!	PRG	1	VECTORS.BOOT	PRG	1	29ERW .ASS	PRG
0	----------------		23	VECTORS.OBJ	PRG	1	30ERW .ASS	PRG
0	---EINGABEHILFEN		0	--------------76		1	21MODUL	SEQ
6	CHECKSUMMER	PRG	1	FARBENSPIEL	PRG	1	22MODUL	SEQ
7	MSE	PRG	0	--------------77		1	23MODUL	SEQ
0	---------C128-22		1	CLIP	PRG	1	24MODUL	SEQ
31	PYRAMIDE	PRG	0	--------------78		1	25MODUL	SEQ
0	--------------36		1	RESET	PRG	1	26MODUL	SEQ
31	PROTERM-64XTW	PRG	0	--------------78		1	27MODUL	SEQ
1	...PARAM	PRG	2	BUNTES LISTING	PRG	1	28MODUL	SEQ
0	--------------52		0	--------------80		1	29MODUL	SEQ
25	R.C.S.	PRG	2	TOKEN-FINDER	PRG	1	30MODUL	SEQ
3	AXEL F.	PRG	0	--------------81		0	-------------133	
11	AXEL F./I	PRG	1	CHESS.BOOT	PRG	3	SIM. C16/C128	PRG
3	NINETEEN	PRG	6	CHESS	PRG	3	SIM. C64/SIM.BAS	PRG
11	NINE/I	PRG	0	---------C128-86		0	-------------142	
7	RPS-DEMO	PRG	2	ZEILEN EINF MOD	PRG	2	SUPER-QUICKSORT	PRG
7	ORIGINAL-RHYTHME	PRG	0	---------C128-86		27	QUICKSORT.ASS	PRG
7	PROGRESS-DEMO	PRG	2	MONITOR MOD	PRG	2	QUICKSORT.COD	PRG
0	--------------56		0	---------C128-87		0	-------------147	
5	BOOT	PRG	2	TRANSFER MOD	PRG	1	GRAF.EINF. FX-80	PRG
25	VARIOPRINT 3.0	PRG	0	--------------89		1	TEXT/GRAF. FX-80	PRG
24	VARIOWRITE 3.0	PRG	9	BIG CHANGE	PRG	1	GRAFIK MPS801	PRG
17	VARIOSET 0 BLOCK	PRG	8	SMALL CHANGE	PRG	1	TEXT/GRAF.MPS801	PRG
17	VARIOSET 1 WRITE	PRG	13	LFNT0	PRG	1	GRAFIK MPS802	PRG
17	VARIOSET 2 DATA	PRG	13	LFNT1	PRG	0	-------------150	
17	VARIOSET 3 NLQ	PRG	13	LFNT2	PRG	2	PG64	PRG
17	VARIOSET 5 OLD	PRG	7	SFNTS	PRG	2	PG64-DEMO	PRG
107	VARIOPRINT 3.1	PRG	0	--------------96		0	-------------168	
0	--------------67		1	21ERW .ASS	PRG	9	ZVIZA 2	PRG
3	UMLAUT2	PRG	1	22ERW .ASS	PRG	1	GABRIELE 9009	PRG
4	MPS802	PRG	1	23ERW .ASS	PRG	4	GAMMA	PRG
4	NORMAL	PRG	1	24ERW .ASS	PRG	2	SEQ-GAMMA	SEQ
4	CENTRONIC	PRG	1	25ERW .ASS	PRG	7	GRIECH./PARALLEL	PRG
6	Z.E.T1	PRG	1	26ERW .ASS	PRG	0	----------------	
11	ZEICH.EDITORV1.0	PRG	1	27ERW .ASS	PRG	0	------ENDE------	
0	---------C128-73		1	28ERW .ASS	PRG	0	----------------	

Bild 1. Ein Beispielausdruck von »Catalog Printer+«

```
5 GOTO 20                                              <191>
10 A$="CATALOG PRINTER+":OPEN 1,8,15,"S:"+
   A$:CLOSE 1:SAVE A$,8:END                            <098>
11 'CAPRI + FILE PRINT ER AUS 64'ER S ON D
   ER2/86 S.111 / 29.5.86 / KD                         <083>
12 'CAPRI BY HERBERT KAUFHOLD, AUF DEN S T
   O ECKEN 17A, 4040 NEUSS 22                          <067>
13 'FILE PRINT ER BY KLAUS GRABIETZ, AM WE
   INBERG 14, 3108 WINSEN                              <147>
14 ' 64 -ERSI ON                                       <184>
15 :                                                   <247>
20 POKE 53280,0:POKE 53281,11:PRINT "{CLR,
   CYAN}"CHR$(14)                                      <110>
30 CLOSE 4:OPEN 4,4,0:CLOSE 4                          <167>
40 IF ST AND-64 THEN PRINT,"{RVSON}DRUCKER
   EINSCHALTEN{UP}":GOTO 30                            <187>
50 :                                                   <026>
60 S=704:DJ=86:REM DJ=JAHR                             <152>
70 C$=CHR$(13):E$=CHR$(27):O$=CHR$(0):GOSU
   B 930:POKE S+3,DJ:POKE 198,0                        <108>
90 :                                                   <066>
100 PRINT"{CLR,2DOWN}"," DIRECTORY PRINTER             <241>
110 PRINT"{9DOWN,3SPACE}NAECHSTE DISKETTE
    -{SPACE,RVSON}'F7'{RVOFF,SPACE}ODER{SP
    ACE,RVSON}RETURN                                   <120>
120 IF W=64 THEN PRINT"{DOWN,3SPACE}DRUCK
    WIEDERHOLEN -{SPACE,RVSON}'F5'"                    <102>
130 PRINT,"{DOWN}VALIDIEREN -{SPACE,RVSON}
    'F3'"                                              <211>
140 PRINT,"{DOWN,6SPACE}ENDE -{SPACE,RVSON
    }'F1'"                                             <140>
150 WAIT 198,1:GET A$:AS=ASC(A$):POKE 198,
    0:IF AS=135 AND W<>64 THEN AS=0                    <200>
160 IF AS=136 AND W=64 THEN GOSUB 300:GOTO
    100                                                <109>
170 IF AS=13 THEN AS=136                               <099>
180 ON AS+132*(AS>132) GOSUB 3000,1020,600
    ,200:GOTO 100                                      <053>
190 :                                                  <166>
200 DJ=PEEK(S+3):DJ$=MID$(STR$(DJ),2)                  <180>
210 GOSUB 800:IF D3$<>DJ$ THEN D$="{RVSON}
    TTMM"+DJ$                                          <148>
220 PRINT"{LIG.GREEN,HOME,6DOWN}TAGESDATUM
    {2SPACE}:"D$C$"{UP}"SPC(11);                       <025>
230 INPUT D$:IF RIGHT$(D$,2)<>DJ$THEN 220              <052>
240 IF D$=DA$ THEN 300                                 <076>
250 POKE S,VAL(MID$(D$,1,2)):POKE S+1,VAL(
    MID$(D$,3,2))                                      <214>
260 POKE S+2,VAL(MID$(D$,5,2)):GOSUB 800:R
    EM DATUM                                           <054>
270 :                                                  <248>
290 REM=== BLOCK'S FREE / DIR-NAME                     <187>
300 W=0:N=0:CC$="":PRINT"{CLR,LIG.GREEN}";
    :GOSUB 1000:IF DE=21 THEN GOSUB 2100:G
    OTO 510                                            <188>
310 OPEN 2,8,2,"#":GET#2,A$:P=ASC(A$+O$)+3             <127>
320 PRINT#15,"M-R"CHR$(250)CHR$(2):GET#15,
    BL$                                                <087>
330 PRINT#15,"M-R"CHR$(252)CHR$(2):GET#15,
    BH$:BF=ASC(BL$+O$)+256*ASC(BH$+O$)                 <037>
340 SP=18:SE=0:GOSUB 870:BF$=RIGHT$("{3SPA
    CE}"+STR$(BF),4)                                   <032>
350 PRINT#15,"M-R"CHR$(144)CHR$(7)CHR$(23)             <250>
360 A$="":CC$="":INPUT#15,A$:CC$=LEFT$(A$,
    16):ID$=MID$(A$,19,2)                              <244>
365 :                                                  <087>
370 SP=18:SE=1:PRINT"{CLR}"," "DD$C$,"{SPA
    CE,RVSON}"CC$"{RVOFF,SPACE,RVSON}"ID$              <100>
380 :                                                  <102>
390 REM=== DIR LESEN                                   <148>
400 GOSUB 870:IF ED=10 THEN 500                        <001>
410 FOR DI=0 TO 7:PZ=DI*32:PRINT#15,"B-P 2
    ";PZ+2:GOSUB 2000:REM ERR                          <073>
420 : IF DE THEN ED=10:DI=10                           <182>
430 : GET#2,A$:A=ASC(A$+O$):B=A-(A AND 128
    ):IF A=0 THEN 490:REM GELOESCHT                    <050>
435 : IF B>64 THEN B=B-60                              <141>
440 : PRINT#15,"M-R"CHR$(PZ+5)CHR$(P)CHR$(
    16):INPUT#15,F$:IF LEN(F$)=16 THEN 470             <223>
450 : PRINT#15,"M-R"CHR$(PZ+5)CHR$(P)CHR$(
    16)                                                <098>
460 : F$="":FOR I=1 TO 16:GET#15,A$:F$=F$+
    A$:NEXT                                            <169>
470 : PRINT#15,"B-P 2";PZ+30:GET#2,BL$:GET
    #2,BH$:BL=ASC(BH$+O$)*256+ASC(BL$+O$)              <027>
480 : N=N+1:F$(N)=RIGHT$("{2SPACE}"+STR$(B
    L),4)+" "+F$+" "+B$(B):PRINT"{6SPACE}"
    F$(N)                                              <088>
490 NEXT:IF ED=0 THEN 400                              <100>
500 PRINT C$"{6SPACE}"BF$" BLOECKE FREI.               <244>
510 CLOSE 15:CLOSE 2                                   <230>
520 :                                                  <244>
590 REM=== PRINT                                       <102>
600 IF CC$="" THEN RETURN                              <196>
610 F=128:OPEN F,4,1:PRINT#F:CMD F:PRINT E
    $"P";                                              <147>
620 PRINT E$"P"E$CHR$(15)E$"S"O$E$"A"CHR$(
    5)E$"-1NAME : "CC$"{2SPACE}ID: "ID$"{2
    SPACE}";                                           <145>
630 PRINT#F,BF$" BLOECKE FREI.{9SPACE}STAN
    D VOM: "DD$" "E$"-0":PRINT#F                       <163>
640 IF W THEN 670                                      <145>
650 B=N/3:B%=B:IF N=0 THEN 710                         <253>
660 IF B<>B% THEN FOR X=1 TO 3:F$(N+X)="":
    NEXT:N=N+3                                         <068>
670 B%=N/3:FOR X=1 TO B%                               <200>
680 : PRINT#F,F$(X)" "F$(X+B%)" "F$(X+2*B%
    )                                                  <005>
690 NEXT                                               <192>
700 :                                                  <168>
710 W=64:CLOSE F:PRINT"{CLR,CYAN}"CHR$(14)
    :RETURN                                            <098>
720 :                                                  <188>
790 REM=== SUBROUTINE                                  <171>
800 D1$=RIGHT$("0"+MID$(STR$(PEEK(S)),2),2
    )                                                  <143>
810 D2$=RIGHT$("0"+MID$(STR$(PEEK(S+1)),2)
    ,2)                                                <232>
820 D3$=RIGHT$("0"+MID$(STR$(PEEK(S+2)),2)
    ,2):DD$=D1$+"."+D2$+"."+D3$                        <143>
830 D$=D1$+D2$+D3$:DA$=D$:RETURN                       <001>
840 :                                                  <054>
850 REM"{2RVSON,SPACE}BLOCK READ                       <225>
860 ED=10:GOTO 900                                     <079>
870 PRINT#15,"U1 2 0";SP;SE:GOSUB 2000:IF
    DE THEN PRINT"ZEILE{SPACE,RVSON}870":G
    OTO 860                                            <038>
880 ED=0:GET#2,SP$:GET#2,SE$:SP=ASC(SP$+O$
    ):SE=ASC(SE$+O$)                                   <174>
890 IF SP=0 OR SP>35 OR SE>20 THEN ED=1                <106>
900 RETURN                                             <196>
910 :                                                  <124>
920 REM"{2RVSON,SPACE}INITIALISIEREN                   <190>
930 J=0:I=0:DI=0:PZ=0:SE=0:A=0:B=0:BL=0:BF
    =0:W=0                                             <078>
940 B$(0)="{4SPACE}":B$(1)="SEQ ":B$(2)="P
    RG ":B$(3)="USR ":B$(4)="REL "                     <031>
950 B$(5)="SEQ<":B$(6)="PRG<":B$(7)="USR<"
    :B$(8)="REL<"                                      <215>
960 DIM F$(152)                                        <195>
970 RETURN                                             <010>
980 :                                                  <194>
990 REM"{2RVSON,SPACE}DISK-ANW.                        <075>
1000 OPEN 15,8,15,"I0":GOSUB 2000:RETURN               <171>
1010 :                                                 <224>
1020 OPEN 15,8,15,"V0":GOSUB 2000:RETURN               <144>
1030 :                                                 <246>
1990 REM"{2RVSON,SPACE}DISK-ERR                        <113>
2000 INPUT#15,DE,DE$,S1,S2                             <051>
2010 IF DE THEN PRINT"{DOWN,RVSON}"DE;DE$,
     S1;S2                                             <063>
2020 RETURN                                            <044>
2030 :                                                 <228>
2100 PRINT,"{DOWN,RVSON}DISK ERROR #"DE;C$
     "{DOWN}WEITER MIT:{3SPACE,DOWN,RVSON}
     TASTE                                             <150>
2110 POKE 198,0:WAIT 198,1:RETURN                      <221>
2990 REM=== ENDE                                       <171>
3000 PRINT"{CLR}":CLOSE 15:END                         <039>
```

Listing zu »Catalog Printer+«.
Bitte beachten Sie die Eingabehinweise auf Seite 97.

Laden und Speichern ohne Kompromisse

Das Programm »Load/Save Plus« zeigt, wie man mit wenigen Bytes das Betriebssystem des C64 sinnvoll erweitert. Das Speichern beliebiger Bereiche und ein »verschobenes« Laden ist nun auch ohne Monitor möglich.

Zur Speicherung von Maschinenprogrammen, Bildschirmmasken oder hochauflösenden Grafiken und ähnlichem mußte bisher oft ein Maschinensprache-Monitor eingesetzt werden. Auch das Laden von Diskette in gewünschte andere Speicherbereiche war bisher nur auf Umwegen möglich. »Load/Save Plus« beseitigt dieses Manko des C64 in nur 74 Bytes. Programme können nun von der Diskette in jeden gewünschten RAM-Bereich geladen und aus nahezu allen Bereichen auf Diskette oder Kassette gespeichert werden.

Bedienungsanleitung

Die Erweiterung kann entweder als Basic-Lader (Listing 1) oder direkt als Maschinenprogramm (Listing 2) mit dem MSE eingegeben und gespeichert werden. Der Basic-Lader speichert die Erweiterung nach RUN automatisch als Maschinenprogramm auf Diskette. Nach dem Laden mit »LOAD "LOAD/SAVE.OBJ",8,1« und der Aktivierung mit SYS 694 stehen folgende Befehle zur Verfügung:

- `SAVE"NAME",ga`
  Normale SAVE-Routine
- `SAVE"NAME",ga,1,START,ENDE`
  Bereich von START bis ENDE speichern
- `LOAD"NAME",ga`
  Normales Basic-LOAD
- `LOAD"NAME",ga,1`
  Absolutes Laden
- `LOAD"NAME",8,1,START`
  Laden an angegebene Adresse (nur Disk)
  (ga = 8/9 oder 10/11 für Diskette beziehungsweise ga = 1 für Kassette)

»START« und »ENDE« können direkt, das heißt mit Hilfe ganzer Zahlen, oder indirekt als Variablen angegeben werden. In der vorliegenden Form ist es jedoch nicht möglich, das RAM zwischen $A000 und $BFFF oder $E000 und $FFFF zu speichern.

Beim Aktivieren der Erweiterung mit SYS 694 werden die im RAM liegenden Vektoren für LOAD und SAVE ($0330/0331 und $0332/0333), die normalerweise auf $F4A5 beziehungsweise $F5ED zeigen, geändert. Die Vektoren zeigen nun auf die Erweiterung. Die Routinen prüfen zunächst, ob beim LOAD- oder SAVE-Befehl zusätzliche Parameter angegeben wurden. Wenn das nicht der Fall ist, wird die normale LOAD- beziehungsweise SAVE-Routine angesprungen. Wenn eine START- beziehungsweise ENDE-Adresse angegeben wurde, werden die Zeiger auf Beginn und Ende in der Zeropage entsprechend gesetzt und danach die LOAD-beziehungsweise SAVE-Routine ausgeführt. Zuletzt sei noch darauf hingewiesen, daß die Erweiterung nach <RUN/STOP+RESTORE> mit SYS 694 erneut aktiviert werden muß.

(Michael Möller/nj)

```
100 REM ********************************     <238>
110 REM * LOAD MIT STARTADRESSE        *     <082>
120 REM * SAVE MIT START-/ENDADRESSE   *     <076>
130 REM * WRITTEN BY MICHAEL MOELLER   *     <139>
140 REM ********************************     <022>
150 :                                        <126>
160 REM * EINLESEN *                         <121>
170 FOR I=694 TO 767                         <246>
180 : READ A:POKE I,A:S=S+A                  <203>
190 NEXT                                     <200>
200 :                                        <176>
210 REM * PRUEFEN *                          <001>
220 P=8494                                   <035>
230 IF S=P THEN 270                          <124>
240 PRINT" DATAFEHLER!":PRINT" SUMME IST:
    "S:PRINT" SUMME SOLL:"P:END              <018>
250 :                                        <226>
260 REM *INITIALISIEREN *                    <021>
270 SYS 694                                  <084>
280 :                                        <002>
290 REM * PROGRAMM SPEICHERN *               <194>
300 SAVE"LOAD/SAVE.OBJ",8,0,694,768          <170>
310 :                                        <032>
320 REM * ERKLAERUNGEN *                     <147>
330 PRINT:PRINT"FOLGENDE BEFEHLE STEHEN ZU
    R VERFUEGUNG:":PRINT                     <117>
340 PRINT"SAVE"CHR$(34)"NAME"CHR$(34)",8"T
    AB(27)"- BASIC-SAVE"                     <043>
350 PRINT"SAVE"CHR$(34)"NAME"CHR$(34)",8,0
    ,START,ENDE{2SPACE}- SAVE FREI"          <042>
360 PRINT:PRINT"LOAD"CHR$(34)"NAME"CHR$(34
    )",8"TAB(27)"- BASIC-LOAD"               <023>
370 PRINT"LOAD"CHR$(34)"NAME"CHR$(34)",8,1
    "TAB(27)"- LOAD ABS."                    <068>
380 PRINT"LOAD"CHR$(34)"NAME"CHR$(34)",8,1
    ,START"TAB(27)"- LOAD FREI"              <090>
390 PRINT:PRINT"FUER >START< UND >ENDE< KO
    ENNEN AUCH"                              <187>
400 PRINT"VARIABLEN STEHEN.":END             <248>
410 :                                        <132>
420 REM * DATAS *                            <253>
430 DATA 169,213,162,2,141,48,3,142,49,3,1
    69,234,162,2,141,50,3,142,51,3,96        <156>
440 DATA 32,253,174,32,138,173,32,247,183,
    96,32,121,0,240,11,32,203,2,132,195      <004>
450 DATA 133,196,169,0,133,185,165,10,76,1
    65,244,32,121,0,240,14,32,203,2,132      <021>
460 DATA 193,133,194,32,203,2,132,174,133,
    175,76,237,245                           <120>
```

Listing 1. Dieser Basic-Lader speichert »Load/Save Plus« automatisch als Maschinenprogramm auf Diskette

```
Name : load/save.obj        02b6 0300
-------------------------------------
02b6 : a9 d5 a2 02 8d 30 03 8e   b6
02be : 31 03 a9 ea a2 02 8d 32   0d
02c6 : 03 8e 33 03 60 20 fd ae   9a
02ce : 20 8a ad 20 f7 b7 60 20   a2
02d6 : 79 00 f0 0b 20 cb 02 84   5e
02de : c3 85 c4 a9 00 85 b9 a5   28
02e6 : 0a 4c a5 f4 20 79 00 f0   ce
02ee : 0e 20 cb 02 84 c1 85 c2   31
02f6 : 20 cb 02 84 ae 85 af 4c   7b
02fe : ed f5 8b e3 83 a4 7c a5   e0
```

Listing 2. Maschinenprogramm »Load/Save Plus«. Bitte verwenden Sie zur Eingabe den MSE auf Seite 100

Schneller geht's kaum

Wie würde Ihnen ein Programm gefallen, das in 90 Sekunden eine Diskette kopiert? Bitte schön, hier ist es:

In unserer hektischen Zeit muß alles möglichst schnell gehen, auch das Kopieren von Disketten. Eines der schnellsten Kopierprogramme finden Sie hier als Listing abgedruckt. Es benötigt, rechnet man die Diskettenwechsel nicht ein, etwa 90 Sekunden für eine Kopie. In einer Stunde kann man damit etwa 30 Disketten kopieren.

Nach dem Laden des Programms und dem Start mit »RUN« erwartet »Track Copy« als erstes die Eingabe des Anfangs- und des Endtracks. Wenn Sie normale Disketten kopieren wollen, geben Sie bitte die Zahlen »01« und »35« ein. Wenn Sie einen Floppy-Speeder besitzen, der vierzigspurig arbeitet, können Sie auch dessen Disketten mit »Track Copy« kopieren, indem Sie als Endtrack »40« vorgeben. Für Einzelfälle kann auch der Track 41 kopiert werden, dieser ist jedoch auf vielen Laufwerken gar nicht mehr ansprechbar.

Während des Kopiervorgangs müssen Sie die Disketten wechseln, wenn Sie vom Programm dazu aufgefordert werden. Nach dem Wechseln der Diskette bestätigen Sie dies bitte mit einem Tastendruck.

Alle weiteren Fragen lassen sich durch Drücken der Tasten <J> für Ja und <N> für Nein beantworten.

Ein Hinweis für technisch Interessierte: Beim Kopieren der Spuren 1 bis 35 müssen Disketten viermal gewechselt werden, weil aus Geschwindigkeitsgründen die GCR-Bytes der Diskette nicht decodiert, sondern direkt zum C 64 gesendet werden.

Ein wichtiger Hinweis zum Schluß: Zwischen Laden und Starten des Programms darf kein Reset am Laufwerk ausgelöst werden, es darf also nicht aus- und wieder eingeschaltet werden.

(Erich Steiner/bs)

```
Name : track-copy           0801 121c
---------------------------------------

0801 : 0b 08 c2 07 9e 32 30 36  4a
0809 : 32 00 00 00 00 4c 91 0d  fe
0811 : 20 c7 08 ca e8 2c 00 dd  1c
0819 : 50 fb 2c 00 dd 30 fb a9  15
0821 : 10 8d 00 dd a0 2b bd 00  0e
0829 : 0f 85 95 a9 10 66 95 6a  f5
0831 : 66 95 6a 4a 4a 8d 00 dd  13
0839 : a9 10 66 95 6a 66 95 6a  3c
0841 : 4a 4a 8d 00 dd a9 10 66  4c
0849 : 95 6a 66 95 6a 4a 4a 8d  9d
0851 : 00 dd a9 10 66 95 6a 66  36
0859 : 95 6a 4a 4a 8d 00 dd 88  61
0861 : f0 04 e8 d0 c1 60 ea ea  48
0869 : ea ea ea a9 00 8d 00 dd  e0
0871 : f0 a2 20 c7 08 ca e8 a9  81
0879 : 0b 8d 00 dd a0 05 ea ea  ba
0881 : 88 d0 fb a9 03 8d 00 dd  fe
0889 : a0 2b ad 00 dd 2a 2a 66  cf
0891 : 95 6a 66 95 ea ea ad 00  64
0899 : dd 2a 2a 66 95 6a 66 95  54
08a1 : ea ea ad 00 dd 2a 2a 66  10
08a9 : 95 6a 66 95 ea ea ad 00  7c
08b1 : dd 2a 2a 66 95 6a 66 95  6c
08b9 : a5 95 49 ff 9d 00 0f 88  a2
08c1 : f0 b4 e8 d0 c5 60 78 a9  f4
08c9 : 0b 8d 00 dd 2c 00 dd 10  b1
08d1 : fb a9 03 8d 00 dd 60 c9  17
08d9 : 24 90 05 a9 11 a2 00 60  a3
08e1 : a2 04 dd f0 08 ca b0 fa  aa
08e9 : bd ed 08 60 11 12 13 15  c3
08f1 : 24 1f 19 12 ad 28 08 85  75
08f9 : 15 ad 29 08 85 16 a9 00  e0
0901 : 8d 28 08 8d 29 08 a2 fd  af
0909 : 20 11 08 a5 15 8d 28 08  d7
0911 : a5 16 8d 29 08 60 a2 00  58
0919 : 20 73 08 ce bf 08 a2 bb  0d
0921 : 20 76 08 ee bf 08 4c de  87
0929 : 09 a2 00 20 11 08 ce 29  66
0931 : 08 a2 bb 20 11 08 ee 29  dd
0939 : 08 60 a5 10 20 d8 08 8a  db
0941 : 0a aa bd 52 09 18 6d e8  33
0949 : 09 e8 bd 52 09 6d e9 09  36
0951 : 60 a9 1b 1f 19 da 17 95  21
0959 : 16 a5 10 85 1a a9 00 85  f1
0961 : fd a5 10 85 fe 20 f5 08  be
0969 : 20 17 0a a5 10 20 d8 08  c2
0971 : 85 13 a9 00 85 14 20 17  92
0979 : 09 e6 14 a5 14 c5 13 d0  0c
0981 : f5 e6 10 a5 10 c5 11 d0  b7
0989 : 03 4c d2 0c 20 3b 09 90  0a
0991 : cc 4c ae 0c ad e8 09 85  02
0999 : 18 ad e9 09 85 19 a9 00  eb
09a1 : 8d 50 0a a9 10 8d 51 0a  d5
09a9 : a9 10 85 fd a5 1a 85 fe  bb
09b1 : 20 6e 0d 20 84 0a a5 1a  b3
09b9 : 20 d8 08 85 13 a9 00 85  82
09c1 : 14 20 49 0a e6 14 a5 14  47
09c9 : c5 13 d0 f5 e6 1a ad 50  a1
09d1 : 0a c5 18 d0 d3 ad 51 0a  e2
09d9 : c5 19 d0 cc 60 a9 30 85  18
09e1 : 01 a2 00 bd bb 0e 9d a9  e1
09e9 : 2b e8 d0 f7 a9 45 18 6d  bb
09f1 : e8 09 8d e8 09 8d 0a 0a  17
09f9 : 90 03 ee e9 09 ad e9 09  bb
0a01 : 8d 0b 0a a2 bb bd 00 0f  b2
0a09 : 9d a9 2a e8 d0 f7 ee e9  7f
0a11 : 09 a9 37 85 01 60 a2 00  0b
0a19 : 20 73 08 ad 00 0f c9 ff  4a
0a21 : d0 03 4c 52 0d ad e8 09  c4
0a29 : 8d 3c 0a ad e9 09 8d 3d  a4
0a31 : 0a a9 30 85 01 a2 00 bd  6d
0a39 : 00 0f 9d 00 10 e8 d0 f7  a4
0a41 : ee e9 09 a9 37 85 01 60  00
0a49 : a9 30 85 01 a2 00 bd a9  00
0a51 : 2b 9d bb 0e e8 d0 f7 a9  44
0a59 : 45 18 6d 50 0a 8d 50 0a  72
0a61 : 8d 72 0a 90 03 ee 51 0a  bd
0a69 : ad 51 0a 8d 73 0a a2 bb  7d
0a71 : bd a9 2a 9d 00 0f e8 d0  ff
0a79 : f7 ee 51 0a a9 37 85 01  e9
0a81 : 4c 2a 09 a9 30 85 01 ad  e8
0a89 : 50 0a 8d 97 0a ad 51 0a  9c
0a91 : 8d 98 0a a2 00 bd 00 10  4f
0a99 : 9d 00 0f e8 d0 f7 ee 51  42
0aa1 : 0a a9 37 85 01 a2 00 4c  bc
0aa9 : 11 08 51 55 45 4c 4c 44  2e
0ab1 : 49 53 4b 45 54 54 45 20  5d
0ab9 : 45 49 4e 4c 45 47 45 4e  00
0ac1 : 20 21 5a 49 45 4c 44 49  8c
0ac9 : 53 4b 45 54 54 45 20 45  18
0ad1 : 49 4e 4c 45 47 45 4e 20  15
0ad9 : 21 4e 4f 43 48 20 45 49  8b
0ae1 : 4e 45 20 44 49 53 4b 45  49
0ae9 : 54 54 45 20 4b 4f 50 49  c0
0af1 : 45 52 45 4e 20 3f 20 20  37
0af9 : 47 4c 45 49 43 48 45 20  ad
0b01 : 42 45 44 49 4e 47 55 4e  31
0b09 : 47 45 4e 20 3f 20 20 49  93
0b11 : 43 48 20 52 45 53 45 54  77
0b19 : 45 20 41 4c 4c 45 53 41  07
0b21 : 4e 46 41 4e 47 53 54 52  b1
0b29 : 41 43 4b 20 28 30 31 2d  06
0b31 : 34 32 29 20 3a 45 4e 44  5c
0b39 : 54 52 41 43 4b 20 28 30  26
0b41 : 31 2d 34 32 29 20 3a 52  7d
0b49 : 49 43 48 54 49 47 45 20  f5
0b51 : 45 49 4e 47 41 42 45 20  33
0b59 : 3f 20 20 ad 11 d0 29 ef  82
0b61 : 8d 11 d0 a0 30 a2 00 ca  6d
0b69 : d0 fd 88 d0 fa 60 ad 11  ff
0b71 : d0 09 10 8d 11 d0 ad 00  ca
0b79 : dd 09 03 8d 00 dd 60 8e  db
0b81 : 8c 0b 8c 8d 0b 8d 93 0b  e9
0b89 : a2 00 bd 0f 0b 20 d2 ff  7a
0b91 : e8 e0 11 d0 f5 60 20 9a  60
0b99 : 0b a9 0d 4c d2 ff a9 20  5a
0ba1 : 4c d2 ff a9 00 85 c6 20  13
0ba9 : e4 ff c9 4a 38 f0 05 c9  fb
0bb1 : 4e 18 d0 f3 60 a9 00 85  1c
0bb9 : c6 20 e4 ff f0 fb 60 8a  4e
0bc1 : 38 e9 30 0a 0a 0a 0a 85  5f
0bc9 : 1b 98 38 e9 30 18 65 1b  0b
0bd1 : a0 00 aa f0 0b f8 38 e9  67
0bd9 : 01 d8 f0 03 c8 d0 f6 c8  63
0be1 : 98 60 a9 30 20 d2 ff 20  f3
0be9 : d2 ff 20 b6 0b c9 0d f0  af
0bf1 : 19 c9 30 90 f5 c9 3a b0  05
0bf9 : f1 aa c6 d3 a4 d3 b1 d1  bf
0c01 : 88 91 d1 8a 20 d2 ff 4c  49
0c09 : eb 0b a4 d3 88 b1 d1 48  0b
0c11 : 88 b1 d1 aa 68 a8 60 a9  dc
0c19 : 16 a2 20 a0 0b 20 80 0b  66
0c21 : a2 03 20 9f 0b ca d0 fa  81
0c29 : 20 e3 0b 8e 40 0b 8c 41  e1
0c31 : 0b 20 c0 0b c9 01 90 04  cd
0c39 : c9 2b 90 0a ea ea ea c6  3c
0c41 : d3 c6 d3 4c 29 0c 85 10  1f
0c49 : 85 1c 20 97 0b a9 12 a2  63
0c51 : 36 a0 0b 20 80 0b a2 07  97
0c59 : 20 9f 0b ca d0 fa 20 e3  92
0c61 : 0b 20 c0 0b c5 10 90 04  35
0c69 : c9 2b 90 0a ea ea ea c6  6c
0c71 : d3 c6 d3 4c 5f 0c 85 11  b4
0c79 : e6 11 85 1d e6 1d 20 97  f4
0c81 : 0b a9 14 a2 48 a0 0b 20  b0
0c89 : 80 0b 20 a4 0b b0 09 20  c6
0c91 : d2 ff 20 97 0b 4c 18 0c  e9
0c99 : 20 d2 ff 20 97 0b a9 18  cf
0ca1 : a2 ab a0 0a 20 80 0b 20  f5
0ca9 : b6 0b 4c 41 0d 58 20 6f  13
0cb1 : 0b 20 97 0b a9 17 a2 c3  79
0cb9 : a0 0a 20 80 0b 20 b6 0b  19
0cc1 : 78 20 5c 0b 20 95 09 58  45
0cc9 : 20 6f 0b 20 97 0b 4c 9f  aa
0cd1 : 0c 58 20 6f 0b 20 97 0b  26
0cd9 : a9 17 a2 c3 a0 0a 20 80  0b
0ce1 : 0b 20 b6 0b 78 20 5c 0b  1b
0ce9 : 20 95 09 58 20 6f 0b 20  0b
0cf1 : 97 0b a9 1f a2 da a0 0a  f4
0cf9 : 20 80 0b 20 a4 0b b0 12  aa
0d01 : 20 d2 ff 20 97 0b a9 11  29
0d09 : a2 0f a0 0b 20 80 0b 4c  87
0d11 : d7 0d 20 d2 ff 20 97 0b  47
0d19 : a9 17 a2 f9 a0 0a 20 80  11
0d21 : 0b 20 a4 0b b0 09 20 d2  40
0d29 : ff 20 97 0b 4c 18 0c 20  76
0d31 : d2 ff 20 97 0b a5 1c 85  57
0d39 : 10 a5 1d 85 11 4c 9f 0c  1e
0d41 : 78 20 5c 0b a9 00 8d e8  e4
0d49 : 09 a9 10 8d e9 09 4c 5a  a9
0d51 : 09 ae e8 09 8e 5f 0d ae  82
0d59 : e9 09 8e 60 0d 8d 00 10  d4
0d61 : ee e8 09 d0 03 ee e9 09  81
0d69 : 68 68 4c 82 09 ad 50 0a  bc
0d71 : 8d 7b 0d ad 51 0a 8d 7c  49
0d79 : 0d ad 00 10 c9 ff f0 03  c5
0d81 : 4c f5 08 ee 50 0a d0 03  46
0d89 : ee 51 0a 68 68 4c cd 09  e2
0d91 : a9 15 8d 18 d0 a9 93 20  14
0d99 : 0b 12 a9 0b 85 d3 a9 2c  6f
0da1 : a2 ab a0 0d 20 80 0b 4c  ad
0da9 : e4 0d 54 52 41 43 4b 20  0f
0db1 : 2d 20 43 4f 50 59 0d 20  ed
0db9 : 20 20 20 20 20 20 20 20  b9
0dc1 : 20 20 31 39 38 36 20 20  5b
0dc9 : 20 20 45 2e 53 54 45 49  90
0dd1 : 4e 45 52 0d 0d 0d a9 30  38
0dd9 : 8d fd 0f a2 fd 20 11 08  b2
0de1 : 4c e2 fc a9 01 a2 08 a0  99
0de9 : 6f 20 ba ff a9 00 20 bd  ad
0df1 : ff 20 c0 ff a9 08 20 b1  ef
0df9 : ff a9 6f 20 93 ff a0 00  68
0e01 : b9 64 0e 20 a8 ff c8 c0  a3
0e09 : 06 d0 f5 a0 00 b9 71 0e  b8
0e11 : 20 a8 ff c8 c0 20 d0 f5  da
0e19 : a9 08 20 ae ff ad 0f 0e  6a
0e21 : 18 69 20 8d 0f 0e 90 03  51
0e29 : ee 10 0e ad 67 0e 18 69  73
```

```
0e31 : 20 8d 67 0e 90 03 ee 68  61
0e39 : 0e c9 a0 d0 b7 ad 68 0e  15
0e41 : c9 06 d0 b0 a9 08 20 b1  16
0e49 : ff a9 6f 20 93 ff a0 00  b8
0e51 : b9 6a 0e 20 a8 ff c8 c0  f6
0e59 : 07 d0 f5 a9 08 20 ae ff  b7
0e61 : 4c 18 0c 4d 2d 57 00 03  fa
0e69 : 20 4d 2d 57 00 00 01 e0  2c
0e71 : 4c e2 04 20 c8 03 20 3c  d1
0e79 : 06 85 23 85 24 20 3d 03  fa
0e81 : 50 fe b8 ad 01 1c c9 52  f1
0e89 : d0 f3 a0 00 20 3d 03 a2  b8
0e91 : 0a 50 fe b8 20 3d 03 50  33
0e99 : fe b8 ad 01 1c 99 00 07  1b
0ea1 : c8 ca d0 f3 c6 23 d0 e4  13
0ea9 : 84 26 4c 6c 05 a9 d0 8d  dd
0eb1 : 05 18 2c 05 18 30 05 a9  d9
0eb9 : 01 4c 2d 06 2c 00 1c 30  80
0ec1 : f1 ad 01 1c b8 60 a2 00  66
0ec9 : 20 7b 03 a9 01 8d 89 03  46
0ed1 : a2 bb 20 7e 03 a9 02 8d  c9
0ed9 : 89 03 60 2c 00 18 10 fb  7a
0ee1 : a9 10 8d 00 18 2c 00 18  09
0ee9 : 30 fb 60 20 6b 03 ca e8  ff
0ef1 : 2c 00 18 10 fb a0 2b bd  12
0ef9 : 00 02 8d 85 00 a9 00 66  28
0f01 : 85 2a 2a 66 85 2a 2a 8d  60
0f09 : 00 18 a9 00 66 85 2a 2a  0f
0f11 : 66 85 2a 2a 8d 00 18 a9  96
0f19 : 00 66 85 2a 2a 66 85 2a  33
0f21 : 2a 8d 00 18 a9 00 66 85  54
0f29 : 2a 2a 66 85 2a 2a 8d 00  dd
0f31 : 18 88 f0 bb e8 d0 c0 60  1a
0f39 : 48 38 e5 22 0a 85 4a f0  33
0f41 : 14 20 02 fa a5 62 c9 05  e4
0f49 : f0 0b a2 00 b1 ff b1 ff  49
0f51 : ca d0 f9 f0 ec ad 0c 1c  c4
0f59 : 29 02 8d 0c 1c 68 85 22  c8
0f61 : a2 04 dd 45 06 ca b0 fa  95
0f69 : 8a 0a 0a 0a 0a 0a 85 27  12
0f71 : ad 00 1c 29 9f 05 27 8d  24
0f79 : 00 1c a5 22 60 20 3d 03  37
0f81 : a0 00 50 fe b8 ad 01 1c  4a
0f89 : 99 00 02 c8 d0 f4 a0 bb  6b
0f91 : 50 fe b8 ad 01 1c 99 00  9b
0f99 : 01 c8 d0 f4 20 56 03 e6  60
0fa1 : 23 a5 23 c5 24 f0 36 a2  00
0fa9 : 05 20 55 04 20 3d 03 a4  d5
0fb1 : 25 a2 0a 50 fe b8 ad 01  22
0fb9 : 1c d9 00 07 d0 ee c8 ca  e0
0fc1 : d0 f1 4c 0d 04 a5 25 18  71
0fc9 : 69 0a c5 26 d0 02 a9 00  31
0fd1 : ca d0 f4 85 25 60 a9 28  3d
0fd9 : 85 25 4c 3c 04 4c e2 04  c2
0fe1 : ca e8 a9 02 8d 00 18 a9  57
0fe9 : 04 2c 00 18 f0 fb a9 00  9c
0ff1 : 8d 00 18 a0 04 88 d0 fd  5c
0ff9 : a0 2b ad 00 18 4a 6a 4a  ac
1001 : 66 85 0a 0a 66 85 ad 00  37
1009 : 18 4a 6a 4a 66 85 0a 0a  f9
1011 : 66 85 ad 00 18 4a 6a 4a  b7
1019 : 66 85 0a 0a 66 85 ad 00  4f
1021 : 18 4a 6a 4a 66 85 0a 0a  11
1029 : 66 85 ad 85 00 9d 00 02  5f
1031 : 88 f0 ae e8 d0 c4 60 20  ef
1039 : 6b 03 4c 70 04 a2 00 20  dd
1041 : c7 04 a2 bb a9 01 8d bf  83
1049 : 04 20 70 04 a9 02 8d bf  5a
1051 : 04 60 a2 fd 20 c7 04 ad  99
1059 : fd 02 c9 00 d0 06 ad fe  bc
1061 : 02 4c 03 03 c9 10 d0 06  17
1069 : ad fe 02 4c 52 06 4c 4a  bb
1071 : 06 a9 ff 8d 03 1c ad 0c  dd
1079 : 1c 29 1f 09 c0 8d 0c 1c  f4
1081 : 60 a9 00 85 28 20 01 05  f8
1089 : a9 55 8d 01 1c a2 03 a0  84
1091 : 00 50 fe b8 88 d0 fa ca  20
1099 : d0 f7 a9 ff 8d 01 1c a2  66
10a1 : 05 50 fe b8 ca d0 fa a4  0d
10a9 : 28 a2 0a 50 fe b8 b9 00  4c
10b1 : 07 8d 01 1c c8 ca d0 f3  51
10b9 : a2 02 84 28 a0 5b a9 55  b9
10c1 : 50 fe b8 8d 01 1c 88 d0  25
10c9 : f7 ca d0 f4 e6 23 a5 23  5c
10d1 : c5 24 d0 c6 20 00 fe a9  07
10d9 : 00 85 23 60 a2 00 a9 07  50
10e1 : 8d 89 03 20 7b 03 a9 02  72
10e9 : 8d 89 03 4c 66 04 20 c8  1e
10f1 : 03 20 3c 06 85 24 a2 00  d8
10f9 : a9 07 8d bf 04 20 c7 04  ea
1101 : a9 02 8d bf 04 a9 28 85  40
1109 : 25 a9 00 85 23 18 a6 24  89
1111 : 69 0a ca d0 fb 85 26 20  11
1119 : 11 05 a2 00 20 c7 04 a9  f9
1121 : 01 8d bf 04 a2 bb 20 c7  71
1129 : 04 a9 02 8d bf 04 20 3d  4b
1131 : 03 a4 25 a2 0a 50 fe b8  b5
1139 : ad 01 1c d9 00 07 d0 ee  02
1141 : c8 ca d0 f1 a2 07 50 fe  82
1149 : b8 ca d0 fa a9 ff 8d 03  d1
1151 : 1c ad 0c 1c 29 1f 09 c0  fc
1159 : 8d 0c 1c a9 ff a2 05 8d  6d
1161 : 01 1c b8 50 fe b8 ca d0  2b
1169 : fa a0 00 b9 00 02 50 fe  3a
1171 : b8 8d 01 1c c8 d0 f4 a0  dc
1179 : bb b9 00 01 50 fe b8 8d  2c
1181 : 01 1c c8 d0 f4 50 fe 20  ea
1189 : 00 fe e6 23 a5 23 c5 24  f9
1191 : d0 03 4c e2 04 a2 05 20  fc
1199 : 55 04 4c aa 05 68 68 a9  e1
11a1 : ff 8d 00 02 a2 00 20 7b  49
11a9 : 03 4c e2 04 c9 24 b0 03  92
11b1 : 4c 4b f2 a9 11 60 2b 1f  94
11b9 : 19 12 a9 12 20 c8 03 4c  75
11c1 : a0 ea 48 a9 10 2c 00 1c  b8
11c9 : f0 03 4c 96 06 ad 00 1c  27
11d1 : 49 08 8d 00 1c a2 00 a0  9a
11d9 : 80 ca d0 fd 88 d0 fa a9  00
11e1 : 10 2c 00 1c f0 e7 ad 00  90
11e9 : 1c 09 08 8d 00 1c a9 10  e5
11f1 : 2c 00 1c d0 fb 2c 00 1c  98
11f9 : f0 fb 48 68 48 68 48 68  c0
1201 : ca d0 f7 88 d0 f4 68 4c  31
1209 : 7e 05 20 d2 ff a9 00 8d  d5
1211 : 20 d0 8d 21 d0 a9 05 8d  aa
1219 : 86 02 60 00 ff 00 ff 00  b8
```

Listing zu Track Copy. Bitte mit dem MSE (Seite 100) eingeben.

Quicksort »par excellence«

Einen neuen Quicksort-Algorithmus möchten wir Ihnen im folgenden vorstellen. Es handelt sich dabei um ein sehr schnelles und äußerst komfortables Werkzeug für Ihre Datenverwaltung

Nachdem wir im Laufe unseres 64'er-Magazins schon eine ganze Menge an Sortierprogrammen vorgestellt haben, wollen wir Sie in unserem Sonderheft mit dem wohl besten dieser Algorithmen verwöhnen.

Dieser Algorithmus ist äußerst schnell. Er sortiert 1000 Elemente in nur drei Sekunden und dürfte damit bei weitem allen Anwendungen genügen. Durch einen Basic-Lader, den Sie in Listing 1 abgedruckt finden, können Sie das Quicksort-Programm ganz einfach in eigene Programme einbinden. Es belegt die Zeilennummern von 10 000 bis 10 082 und wird mit

GOSUB 10000

gestartet. Nach einer kurzen Wartezeit von etwa acht Sekunden steht das Quicksort als Maschinenprogramm ab $C000 bis $C209 zur Verfügung und kann vom Basic-Programm aus angesprungen werden.

Die Startadresse von Quicksort ist normalerweise 49164 ($C00C), wobei ein Feld folgendermaßen sortiert werden kann:

Zum Beispiel das Feld A mit der Dimensionierung 100:

SYS (49164), A(1), A(100)

Wie Sie aus dieser Syntax erkennen können, kann sowohl der Name als auch die Dimensionierung beliebig sein. In unseren früher vorgestellten Algorithmen mußte das zu sortierende Feld jeweils das erste dimensionierte Feld in einem Programm sein. Bei dem neuen Quicksort-Programm spielt das keine Rolle mehr. Sie können jedes Feld sortieren. Dabei ist es egal, ob es sich um ein String-, Integer- oder Real-Variablenfeld handelt.

Auch die Grenzen der Sortierung können Sie angeben. Dabei bleibt der Rest des Feldes unberührt. Tippen Sie zum Beispiel

SYS (49164), A(20), A(30)

so werden nur die Elemente A(20) bis einschließlich A(30) sortiert.

Um den »Hardware-Stack« nicht zu belasten, werden hinter dem Programm 134 Byte für einen »Software-Stack« verwendet. Für die Assembler-Programmierer unter Ihnen zeigt Bild 1 noch einmal die Dokumentation der einzelnen Routinen im Quicksort-Programm. (U. Weingärtner/ks)

```
SYNTAX: SYS 49164, (linkes Grenzelement),
                   (rechtes Grenzelement)
$C000   Dimensionierte Variablen holen
        C000    Variablenadresse holen
        C003    Mit Anfangsadresse DIM's
                vergleichen
        C009    kleiner?
$C00C   Einsprung (Parameterauswertung)
        C000    Anfangsadresse holen
```

Bild 1. Adressenbelegung bei Quicksort

```
          C016   Name merken
          C01C   Endadresse holen
          C023   Variablennamen vergleichen
          C02D   Endadresse größer Anfangsadresse?
          C033   gleich?
          C035   größer?
          C03B   Adresse merken
          C03F   Variablenlänge holen
                 (INT=2/TEXT=3/REAL=5)
          C051   Stack initialisieren
$C056     QUICKSORT: $FB/FC linkes Element (LI)
                     $FD/FE rechtes Element (RE)
                     $57 Variablenlänge
          C056   58/59 = Endadresse - Anfangsadresse
          C063   58/59 = INT (58/59 durch 57)
          C07D   X-Reg/Y-Reg = 58/59 mal 57
          C092   58/59 = X-Reg/Y-Reg + Anfangs-
                 adresse
          C09C   Mittleres Element der Liste (P)
                 nach $62 bis $66
          C0A6   58/59 (X) = LI: 5A/5B (Y) = RE
          C0AF   Vergleiche X mit Y
          C0B9   größer gleich?
          C0BB   drehe Elemente um P
          C0C1   Vergleiche LI mit Y
          C0CB   größer gleich?
          C0CD   RE u. X auf Stack
          C0E9   sortiere Teilliste links (LI bis Y)
          C0EC   RE u. X zurück
          C100   Vergleiche X mit RE
          C10A   größer gleich?
          C114   sortiere Teilliste rechts
                 (X bis RE)
$C117     Accu auf programmierten Stack
$C126     Accu vom programmierten Stack
$C130     drehe Elemente um P:
                 Alle Elemente kleiner P nach links
                 Alle Elemente größer P nach rechts
          C130   Y merken
          C136   Y = Zeiger (P)
          C13E   Vergleiche Wert (X) mit Wert (Y)
          C141   größer gleich?
          C143   X = X+1
          C149   Y zurück
          C14F   X merken
          C155   X = Zeiger (P)
          C15D   Vergleiche Wert (X) mit Wert (Y)
          C160   größer gleich
          C162   Y = Y-1
          C168   X zurück
          C16E   Vergleiche Y mit X
          C178   kleiner?
          C17A   gleich?
          C17C   tausche Wert (X) mit Wert (Y)
          C18C   X = X+1
$C18F     Y = Y-1
$C19B     X = X+1
$C1A7     Vergleiche numerische Werte
$C1B5     Vergleiche REAL-Werte
$C1CE     Vergleiche INT-Werte
$C1DA     Vergleiche Texte
$C1FE     Vergleiche Wert (X) mit Wert (Y)
```

Bild 1. Dokumentation des Quicksort-Programms von Listing 1

```
10000 AN=49152:AD=AN:EN=AN+522                    <081>
10002 PS=0:FOR I=0 TO 15:READ X:PS=PS+X:PO
      KE AD+I,X:NEXT:READ X:AD=AD+16               <022>
10004 IF PS<>X THEN SYS 45640                      <167>
10006 IF AD<EN THEN 10002                          <097>
10008 FOR I=0 TO 21:READ AD:AD=AD+AN:X=PEE
      K(AD+1)*256+PEEK(AD)+AN                      <237>
10010 Y=INT(X/256):X=X-Y*256:POKE AD,X:POK
      E AD+1,Y:NEXT                                <190>
10012 QS=AN+12:RETURN                              <117>
10014 DATA 32,139,176,196,48,208,2,197,47,
      144,44,96,32,253,174,32,1820                 <052>
10016 DATA 0,0,133,251,132,252,165,69,72,1
      65,70,72,32,253,174,32,1872                  <096>
10018 DATA 0,0,170,104,197,70,208,15,104,1
      97,69,208,10,196,252,208,2008                <238>
10020 DATA 2,228,251,240,5,176,4,76,72,178
      ,96,134,253,132,254,36,2137                  <200>
10022 DATA 13,48,10,36,14,48,3,169,5,44,16
      9,2,44,169,3,133,910                         <097>
10024 DATA 87,169,0,141,15,194,165,253,56,
      229,251,133,88,165,254,229,2429              <019>
10026 DATA 252,133,89,162,15,169,0,24,38,8
      8,38,89,42,176,4,197,1516                    <185>
10028 DATA 87,144,3,229,87,56,202,208,239,
      38,88,38,89,162,0,160,1830                   <234>
10030 DATA 0,165,87,133,90,24,138,101,88,1
      70,152,101,89,168,198,90,1794                <094>
10032 DATA 208,244,138,101,251,133,88,152,
      101,252,133,89,160,4,177,88,2319             <114>
10034 DATA 153,98,0,136,16,248,162,3,181,2
      51,149,88,202,16,249,165,2117                <120>
10036 DATA 89,197,91,208,4,165,88,197,90,1
      76,6,32,48,1,76,175,1643                     <201>
10038 DATA 0,165,252,197,91,208,4,165,251,
      197,90,176,51,165,253,32,2297                <188>
10040 DATA 23,1,165,254,32,23,1,165,88,32,
      23,1,165,89,32,23,1117                       <080>
10042 DATA 1,165,90,133,253,165,91,133,254
      ,32,86,0,32,38,1,133,1607                    <111>
10044 DATA 89,32,38,1,133,88,32,38,1,133,2
      54,32,38,1,133,253,1296                      <000>
10046 DATA 165,89,197,254,208,4,165,88,197
      ,253,176,22,165,88,133,251,2455              <017>
10048 DATA 165,89,133,252,76,86,0,174,15,1
      94,48,7,157,16,194,238,1844                  <134>
10050 DATA 15,194,96,76,53,164,206,15,194,
      174,15,194,189,16,194,96,1891                <029>
10052 DATA 165,90,72,165,91,72,169,98,133,
      90,169,0,133,91,32,254,1824                  <167>
10054 DATA 1,176,6,32,155,1,76,62,1,104,13
      3,91,104,133,90,165,1330                     <228>
10056 DATA 88,72,165,89,72,169,98,133,88,1
      69,0,133,89,32,254,1,1652                    <252>
10058 DATA 176,6,32,143,1,76,93,1,104,133,
      89,104,133,88,165,91,1435                    <222>
10060 DATA 197,89,208,4,165,90,197,88,144,
      32,240,16,164,87,136,177,2034                <026>
10062 DATA 88,170,177,90,145,88,138,145,90
      ,136,16,243,32,155,1,165,1879                <173>
10064 DATA 90,56,229,87,133,90,176,2,198,9
      1,96,165,88,24,101,87,1713                   <101>
10066 DATA 133,88,144,2,230,89,96,160,0,17
      7,88,209,90,208,5,200,1919                   <030>
10068 DATA 196,87,208,245,96,160,1,177,88,
      48,6,177,90,16,232,56,1883                   <160>
10070 DATA 96,177,90,16,7,32,167,1,240,3,1
      44,243,24,96,160,0,1496                      <121>
10072 DATA 177,88,16,231,177,90,48,207,24,
      96,160,2,177,88,153,92,1826                  <195>
10074 DATA 0,177,90,153,95,0,136,16,243,20
      0,196,92,240,11,196,95,1940                  <044>
10076 DATA 240,7,177,93,209,96,240,241,96,
      165,92,197,95,96,165,87,2296                 <103>
10078 DATA 201,2,240,202,201,3,240,210,208
      ,171,0,0,0,0,0,0,1678                        <054>
10080 DATA 16,32,188,191,208,213,218,223,2
      34,237,242,247,252,277,319,324,327           <193>
10082 DATA 350,355,358,397,454                     <011>
```

Listing 1. Basic-Lader für Quicksort. Bitte verwenden Sie zur Eingabe den »Checksummer« (Hinweise auf Seite 97).

Daten individuell und professionell verwaltet

Werfen Sie Ihre Zettelkartei aus dem Fenster! Diese altertümliche Methode, Ordnung in eine Sammlung zu bringen, gehört nun dank »DATEV« endgültig der Vergangenheit an.

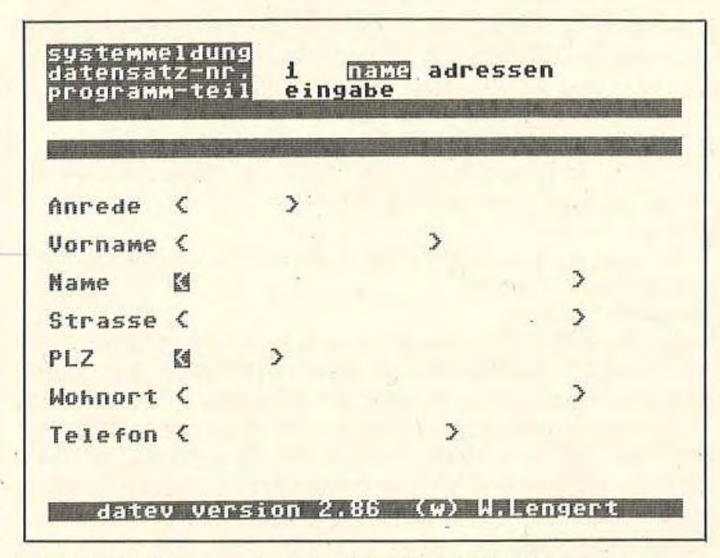


Bild 1. Beispiel einer fertigen Adreßdatei-Eingabemaske

Das universelle Dateiverwaltungsprogramm »DATEV« (Listing 1) ist vollständig in Assembler geschrieben und besitzt viele Funktionen, die man eigentlich nur von einem teuren professionellen Programm erwarten würde. Es ermöglicht die Verwaltung von bis zu 1024 Datensätzen bei einer maximalen Datensatzlänge von 256 Zeichen. Bis zu drei von maximal 15 Feldern können hierbei jeweils als Indexfelder definiert werden (Indexfelder erlauben den schnellen, direkten Zugriff auf einen Datensatz). Systematisches Durchsuchen der Einträge nach bestimmten Kriterienkombinationen ist ebenfalls über die Indexfelder möglich. Die gespeicherten Datensätze können auf Wunsch nach diesen Feldern sortiert werden. Die Form der Ausgabe auf dem Bildschirm oder einen Drucker kann über eine spezielle Formatzeile definiert werden. Auf diese Weise stellt auch das Drucken von Etiketten kein Problem dar.

Bedienungshinweise

Das Programm wird mit »LOAD "DATEV",8« geladen und mit »RUN« gestartet. Nach dem Start erscheint zunächst eine Übersicht der Kommandos und nach einem Tastendruck das Hauptmenü. Durch Drücken der Funktionstasten erreichen Sie folgende Routinen:

<F1> Laden einer Datei von Diskette
<F2> Senden von Befehlen zur VC 1541
<F3> Inhaltsverzeichnis der Diskette
<F5> Neue Datei anlegen
<RUN/STOP> Programmende

Wurde zuvor eine Datei geladen, zusätzlich noch:
<F7> Geladene Datei aktivieren

Neue Datei anlegen (<F5>):

Wenn Sie diesen Programmteil anwählen, ist es wichtig, daß Sie sich über die Anzahl und Länge der einzelnen Felder im klaren sind, da ein Ändern der Eingabemaske nach dem Eröffnen der Datei nicht mehr möglich ist. Eine große Rolle spielen dabei auch die Indexfelder (siehe auch unter Programmteil »Suchen«).

Geben Sie nun Ihre Maske ein. Dabei wird der Anfang eines Feldes mit '<' und das Ende mit '>' gekennzeichnet. Indexfelder werden am Anfang mit einem reversen '<' gekennzeichnet. Als Beispiel soll die Maske einer Adreßdatei dienen (Bild 1).

Der Einfachheit halber sind die Zeichen zur Kennzeichnung der Felder auch auf die Funktionstasten <F1>, <F3> und <F5> gelegt. Nachdem Sie die Maske erstellt haben, speichern Sie diese ab. Drücken Sie dazu <F7>, und geben Sie dann einen Dateinamen ein. Nach dem Abschluß der Eingabe mit <RETURN> wird die Datei auf der Diskette angelegt. Dieser Vorgang kann je nach Größe der Datei, bedingt durch die geringe Geschwindigkeit der Floppy 1541, bis zu einigen Minuten dauern. Danach befinden Sie sich wieder im Auswahlmenü.

Nach dem Öffnen der Datei sind die ersten 256 Datensätze freigegeben. Sind diese editiert, wird Platz für weitere 256 Datensätze geschaffen. Wurden bereits 1024 Datensätze eingegeben, so erscheint die Systemmeldung »Datei voll«.

Datei laden (<F1>):

Zur Eingabe der Daten muß die Datei zuvor mit diesem Programmpunkt geladen werden. Drücken Sie die Taste <F1>, und geben Sie den Namen der Datei ein. Dieser kann mit »*« abgekürzt werden. Wenn Sie anstelle eines Namens nur <RETURN> drücken oder auch nach einer Fehlermeldung, gelangen Sie wieder ins Auswahlmenü. Nachdem die Datei geladen ist, befinden Sie sich im Eingabemodus.

Dateneingabe:

Die Eingabe eines jeden Feldes wird mit <RETURN> abgeschlossen. Ist das letzte Feld erreicht, wird der Datensatz auf Diskette gespeichert und der nächste Datensatz kann eingegeben werden.

Durch Drücken der <F1>-Taste wird die letzte Eingabe des Feldes wieder sichtbar.

Kommandos:

Mit der <CBM>-Taste erreichen Sie die Kommandoeingabe. Folgende Befehle stehen zur Verfügung:

(ungeSHIFTet)
<l> = Löschen
<a> = Ändern
<d> = Drucken
<s> = Suchen
<f> = Formatieren (Ausgabe Drucker)
<e> = Eingabe
<q> = Quit (Programmende)
<c> = Color (Farben ändern)
<w> = Wählen
<+> = Blättern (vor)
<-> = Blättern (zurück)
<h> = Hardcopy
<z> = Zeichensatz wechseln
<*> = Alphabetische Ausgabe
(geSHIFTet)
<A> = Ändern (global)
<D> = Drucken (global)
<M> = Monitor (global)
<L> = Löschen (global)

Bei Dateien ohne Indexfelder entfällt das Kommando »s« (Suchen) und somit auch die Kommandos »l« (Löschen), »a« (Ändern) und »m« (Monitor). Das Kommando »d« (Drucken) bewirkt einen Ausdruck der gesamten Datei.

Löschen (<L>):
Nach Bestätigen der Sicherheitsabfrage mit »J« für Ja wird der angezeigte Datensatz gelöscht.

Ändern (<A>):
Der gerade angezeigte Datensatz kann durch Überschreiben geändert werden.

Drucken (<D>):
Der angezeigte Datensatz wird ausgedruckt (siehe auch Programmteil »Formatieren«). Datev spricht in der abgedruckten Form einen Drucker mit Centronics-Interface am User-Port an. Soll ein serieller Drucker angeschlossen werden, so muß vor dem Starten des Programms »POKE 8613,0« eingegeben und die geänderte Version danach mit »SAVE"filename",8« gespeichert werden.

Eingabe (<E>):
Mit dieser Taste verlassen Sie den Kommandomodus und kehren zurück zur Dateneingabe.

Suchen (<S>):
Dieses Kommando ist nur möglich, wenn Indexfelder definiert wurden. Der Cursor befindet sich im ersten Indexfeld, und Sie können den ersten Suchbegriff eingeben (maximal 8 Zeichen). Die Eingabe wird mit <RETURN> beendet, worauf der Cursor ins nächste Indexfeld springt. Sie können Indexfelder leer lassen, indem Sie nur <RETURN> drücken. Nach der letzten Eingabe wird die Suche gestartet. Wurden Datensätze gefunden, kann die Ausgabe wahlweise über Drucker oder Monitor erfolgen. Die Suchfunktion arbeitet folgendermaßen: Jedes Indexfeld wird nacheinander mit den im Speicher stehenden Daten verglichen. Dabei werden als gefunden in die Tabelle eingetragen:

- Indexfelder, die leer sind (kein Suchbegriff eingegeben)
- Indexfelder mit Übereinstimmung der ersten acht Zeichen
- Indexfelder, die bis zum Leerzeichen übereinstimmen.

Haben Sie beispielsweise eine Adreßkartei mit den Indexfeldern NAME und PLZ angelegt und möchten nun alle »Müllers« in »Düsseldorf« suchen, geben Sie im Indexfeld NAME »Müller« und im Indexfeld PLZ »4000« ein. Es werden nun alle Müllers, die in Düsseldorf wohnen, in die Tabelle eingetragen. Geben Sie bei PLZ nur »4« ein, so werden alle Adressen mit der Postleitzahl 4xxx eingetragen (x steht hier für beliebige Zeichen). Dort stehen sie dann zur Ausgabe zur Verfügung. Wenn Sie alle Indexfelder leer lassen (nur <RETURN> drücken), wird die gesamte Datei in die Tabelle eingetragen und kann dann zum Beispiel ausgedruckt werden.

Wählen (<W>):
Durch Eingabe einer zulässigen Datensatznummer kann jeder Datensatz direkt angewählt und angezeigt werden.

Farben ändern (<C>):
Mit den Funktionstasten <F1>, <F3>, <F5> und <F7> werden die Farben geändert. <RETURN> führt wieder zur Kommandoeingabe.

Quit (<Q>):
Das Programm wird beendet und alle Änderungen auf Diskette gespeichert. Es ist empfehlenswert, die Datei nach längerem Arbeiten zu speichern. Vom Auswahlmenü können Sie die Datei durch Drücken der Taste <F7> wieder aktivieren.

Formatieren (<F>):
Im Eingabefeld erscheint eine Leerzeile mit einem abschließenden »:«. Über diese Zeile wird der Ausdruck gesteuert. Hierbei bedeuten:

»:« Ende der Formatzeile
»0« bis »9« Feldnummern
»;« Leerfeld
»,« Zeilenvorschub
»*« Datensatznummer

Die Feldnummern werden von links nach rechts und von oben nach unten festgelegt. Die Ziffern können auch bei der Maskeneingabe vor die Felder geschrieben werden.

Nehmen wir als Beispiel unsere Adreßdatei. Wollen Sie zum Beispiel Adressen auf Etiketten drucken, könnte die Formatzeile wie folgt aussehen:
»1,2,3,,4;;5,:«

Als erstes wird die Anrede ausgedruckt, dann ein Zeilenvorschub, danach Name/Vorname, wieder ein Zeilenvorschub, nun die Straße, jetzt zwei Zeilenvorschübe, dann die PLZ gefolgt von zwei Leerzeichen, daneben der Ort, und zum Abschluß noch ein Zeilenvorschub. Der Doppelpunkt kennzeichnet das Ende. Durch Ändern der Formatzeile kann die Druckausgabe jederzeit neu formatiert werden. Es ist unbedingt notwendig, als letztes einen Zeilenvorschub (»,«) in die Formatzeile einzugeben.

Blättern (<+>/<->):
Der nächste beziehungsweise vorhergehende Datensatz wird angezeigt. Mit diesem Kommando können Sie sämtliche Datensätze in der eingegebenen Reihenfolge durchblättern. Mit <RETURN> kommen Sie zurück zur Kommandoeingabe.

Hardcopy (<H>):
Nach Drücken dieser Taste wird eine Kopie des Bildschirms auf dem Drucker ausgegeben.

Zeichensatz laden (<Z>):
Das Programm lädt einen Zeichensatz nach, wenn er sich unter dem Namen »ZEICHENSATZ 3« auf der Diskette befindet. Der Zeichensatz muß den Speicherplatz von $0800 bis $1000 belegen. Diese Routine wird nach dem Start von »DATEV« automatisch ausgeführt.

Alphabetische Ausgabe (<*>):
In das Indexfeld, nach dem der alphabetische Ausdruck erfolgen soll, geben Sie bitte den »Pfeil nach links« ein. Die restlichen Indexfelder können wie bei der Funktion »Suchen« benutzt werden.

Globale Kommandos

Die geSHIFTeten Kommandos »L«, »D«, »M« und »A« beziehen sich immer auf die von der Suchroutine erstellte Tabelle. Das heißt, alle in dieser Tabelle eingetragenen Datensätze werden von diesen Kommandos beeinflußt.

Global Drucken (<SHIFT><D>):
Alle Datensätze der Tabelle werden nach Vorgabe der Formatzeile auf dem Drucker ausgegeben. Die Ausgabe kann mit <RUN/STOP> gestoppt, mit <RETURN> fortgesetzt, beziehungsweise mit der <CBM>-Taste abgebrochen werden.

Global Löschen (<SHIFT><L>):
Alle Datensätze in der Tabelle werden nach Bestätigung der Sicherheitsabfrage mit »J« gelöscht.

Monitor (<SHIFT><M>):
Hier gilt das gleiche wie bei »Drucken«, die Ausgabe erfolgt jedoch nun auf den Bildschirm.

Global Ändern (<SHIFT><A>):
Alle Datensätze in der Tabelle können mit diesem Befehl gleichzeitig geändert werden. Felder, die nicht geändert werden sollen, überspringen Sie einfach mit <RETURN>. In die Felder, die global geändert werden sollen, geben Sie den neuen Eintrag ein. Nach Beendigung der Eingabe werden alle Datensätze in der Tabelle auf die gleiche Weise geändert.

System-Fehlermeldungen

Die im folgenden beschriebenen Meldungen werden blinkend angezeigt. Nach <RETURN> werden sie wieder gelöscht.

»Eingabe nicht erlaubt«:
Es wurde ein falsches Kommando eingegeben. Geben Sie erneut einen Befehl ein.
»Feldlänge zu groß«:
Ein Feld übersteigt die maximale Länge von 255 Zeichen. Ändern Sie die Maske entsprechend.
»Max. 15 Felder«:
Es wurden mehr als 15 Felder definiert. Ändern Sie die Maske.
»Max. 3 Indexfelder«:
Es wurden mehr als 3 Indexfelder definiert. Die Maske muß korrigiert werden.
»Datensatz zu lang«:
Alle Felder zusammen übersteigen die maximale Länge von 255 Zeichen. Kürzen Sie die Feldlängen.
»Datensatz-Nr. zu hoch«:
Es wurde eine Datensatz-Nummer angesprochen, die nicht existiert. Wählen Sie eine kleinere Nummer.
»Bitte eine Zahl eingeben«:
Es wurden unerlaubte Zeichen eingegeben. Der Computer verlangt jedoch eine Zahl.
»Ausgabe Formatieren«:
Sie wollten etwas ausdrucken und haben die Formatzeile noch nicht eingegeben.
»Datentabelle leer«:
Sie haben ein globales Kommando eingegeben, jedoch über die Suchfunktion die Tabelle noch nicht erstellt.
»Datei voll«:
Die maximale Anzahl von 1024 Datensätzen ist erreicht. Löschen Sie, wenn möglich, unwichtige Datensätze.
»Falscher Dateiname«:
Sie haben einen Namen eingegeben, der auf der Diskette existiert, jedoch keine Datei ist.
»Datensatz nicht gefunden«:
Es wurde kein Datensatz mit dem(n) angegebenen Suchbegriff(en) gefunden.

Hinweise zum Programm

Geben Sie das Listing bitte mit dem MSE ein und speichern es auf Diskette. Das Programm belegt 32 Blöcke und im Speicher den Bereich von $8000 bis $9F84. Die Indexfelder werden im RAM von $2000 bis $7FFF abgelegt. Der Bereich von $C000 bis $CF00 beinhaltet die Maske, die Parameter, und wird zur Zwischenspeicherung benutzt. Das Programm läuft zusammen mit »HYPRA-LOAD«. Die Funktionstasten werden durch Einschreiben des Wertes 32 in die Speicherstelle $02 abgeschaltet. Wurde DATEV versehentlich mit <RUN/STOP> verlassen, kann es mit »SYS 32768« erneut gestartet werden.

(Wilfried Lengert/nj)

```
Name : datev                    0801 27b0
-----------------------------------------
0801 : 0b 08 c1 07 9e 32 30 36   0a
0809 : 31 00 00 00 a9 2c a0 08   c9
0811 : 85 5f 84 60 a9 b0 a0 27   64
0819 : 85 5a 84 5b a9 84 a0 9f   d8
0821 : 85 58 84 59 20 bf a3 4c   46
0829 : 00 80 00 a9 00 8d 20 80   8d
0831 : 20 af 8f a9 08 20 d2 ff   0f
0839 : 20 5c 99 4c c7 8f 00 00   70
0841 : 00 00 00 00 00 00 00 00   42
0849 : 00 00 00 00 00 01 02 04   62
0851 : 08 10 20 40 80 3b 2c 3a   78
0859 : 20 2a 31 32 33 34 35 36   37
0861 : 37 38 39 30 00 40 60 80   8d
0869 : 05 0d 07 02 00 00 00 00   f7
0871 : 00 00 00 00 00 00 00 00   72
0879 : 20 00 00 08 10 00 00 00   9c
0881 : 00 00 00 ff 00 00 00 00   81
0889 : 05 03 00 00 00 00 04 00   20
0891 : 01 01 08 02 04 04 07 00   d2
0899 : 50 65 01 00 01 0d 00 00   55
08a1 : 00 00 00 00 00 00 00 00   a2
08a9 : 00 00 00 00 00 00 00 00   aa
08b1 : 00 00 00 00 00 00 00 00   b2
08b9 : 00 00 00 00 00 00 00 00   ba
08c1 : 00 00 00 00 00 00 00 00   c2
08c9 : 00 00 00 00 00 00 00 00   ca
08d1 : 00 00 00 00 00 00 00 00   d2
08d9 : 00 00 53 3a 00 00 00 00   f6
08e1 : 00 00 00 00 00 00 00 00   e2
08e9 : 00 00 00 00 00 00 00 00   ea
08f1 : 00 00 00 00 00 00 00 00   f2
08f9 : 00 00 00 00 00 00 00 00   fa
0901 : 00 00 00 00 00 00 00 2e   5e
0909 : 52 45 4c 2c 4c 2c 00 00   bd
0911 : 00 20 20 20 20 20 20 20   f1
0919 : 20 20 20 20 20 20 20 20   19
0921 : 20 20 20 20 20 20 20 20   21
0929 : 20 20 20 20 20 20 20 20   29
0931 : 20 20 20 20 20 20 20 3a   65
0939 : 00 20 46 45 4c 44 4c 41   1e
0941 : 45 4e 47 45 20 5a 55 20   92
0949 : 47 52 4f 53 53 20 00 20   6e
0951 : 4d 41 58 2e 20 31 35 20   bb
0959 : 46 45 4c 44 45 52 20 20   85
0961 : 20 20 20 20 00 20 44 41   32
0969 : 54 45 4e 53 41 54 5a 20   be
0971 : 5a 55 20 4c 41 4e 47 20   eb
0979 : 20 20 00 20 4d 41 58 2e   4a
0981 : 20 33 20 49 4e 44 45 58   39
0989 : 20 46 45 4c 44 45 52 20   9f
0991 : 00 20 44 41 54 45 4e 53   2a
0999 : 41 54 5a 2d 4e 52 2e 20   b1
09a1 : 5a 55 20 48 4f 43 48 20   27
09a9 : 00 20 44 41 54 45 4e 53   42
09b1 : 41 54 5a 20 4e 49 43 48   84
09b9 : 54 20 47 45 46 55 4e 44   69
09c1 : 45 4e 20 00 42 49 54 54   9e
09c9 : 45 20 45 49 4e 45 20 5a   dd
09d1 : 41 48 4c 20 45 49 4e 47   b4
09d9 : 45 42 45 4e 21 20 00 20   ae
09e1 : 41 55 53 47 41 42 45 20   06
09e9 : 46 4f 52 4d 41 54 49 45   7b
09f1 : 52 45 4e 21 20 00 20 44   a9
09f9 : 52 55 43 4b 45 52 20 4e   34
0a01 : 49 43 48 54 20 42 45 52   56
0a09 : 45 49 54 20 00 20 44 41   a1
0a11 : 54 45 4e 54 41 42 45 4c   fa
0a19 : 4c 45 20 4c 45 45 52 21   a4
0a21 : 20 00 20 44 49 53 4b 45   b9
0a29 : 54 54 45 20 4e 49 43 48   ca
0a31 : 54 20 42 45 52 45 49 54   ec
0a39 : 20 00 3e 3e 20 45 49 4e   9f
0a41 : 47 41 42 45 20 4e 49 43   82
0a49 : 48 54 20 45 52 4c 41 55   a3
0a51 : 42 54 3c 3c 00 41 55 53   5a
0a59 : 47 41 42 45 a0 41 55 46   70
0a61 : a0 3e 4d 3c 4f 4e 49 54   30
0a69 : 4f 52 20 4f 44 45 52 20   cb
0a71 : 3e 44 3c 52 55 43 4b 45   52
0a79 : 52 20 00 12 20 46 31 20   57
0a81 : 92 20 3c 20 12 20 46 33   d8
0a89 : 20 92 20 3e 20 12 20 46   62
0a91 : 35 20 92 20 12 3c 92 20   0d
0a99 : 12 20 46 37 20 92 20 53   f2
0aa1 : 50 45 49 20 12 20 46 38   96
0aa9 : 20 92 20 4d 45 4e 55 00   e0
0ab1 : 20 44 41 54 45 49 20 56   9a
0ab9 : 4f 4c 4c 20 00 12 20 20   97
0ac1 : 20 44 41 54 45 56 20 56   12
0ac9 : 45 52 53 49 4f 4e 20 32   82
0ad1 : 2e 38 36 20 20 28 57 29   a0
0ad9 : 20 d7 2e cc 45 4e 47 45   78
0ae1 : 52 54 20 20 20 20 20 92   12
0ae9 : 00 12 20 46 31 20 92 20   62
0af1 : 20 20 44 41 54 45 49 20   2f
0af9 : 56 4f 4e 20 44 49 53 4b   01
0b01 : 45 54 54 45 20 4c 41 44   20
0b09 : 45 4e 0d 0d 12 20 46 32   fa
0b11 : 20 92 20 20 20 44 49 53   76
0b19 : 4b 45 54 54 45 4e 20 42   72
0b21 : 45 46 45 48 4c 45 0d 0d   21
0b29 : 12 20 46 33 20 92 20 20   9b
0b31 : 20 49 4e 48 41 4c 54 53   01
0b39 : 56 45 52 5a 45 49 43 48   4e
0b41 : 4e 49 53 20 44 45 52 20   05
0b49 : 44 49 53 4b 45 54 54 45   43
0b51 : 0d 0d 12 20 46 35 20 92   21
0b59 : 20 20 20 4e 45 55 45 20   b0
0b61 : 44 41 54 45 49 20 45 52   53
0b69 : 4f 45 46 46 4e 45 4e 0d   18
0b71 : 0d 0d 0d 0d 20 20 20 20   ae
0b79 : 20 20 20 50 52 4f 47 52   1d
0b81 : 41 4d 4d 20 45 4e 44 45   23
0b89 : 20 20 00 4b 4f 4d 4d 41   3a
0b91 : 4e 44 4f 20 3f 20 00 57   7d
0b99 : 45 49 54 45 52 21 20 3e   6c
0ba1 : d2 c5 d4 d5 d2 ce 3c 20   1a
0ba9 : 2f 20 41 42 42 52 55 43   14
0bb1 : 48 21 20 3e c3 c2 cd 3c   5c
0bb9 : 20 00 12 53 59 53 54 45   d4
0bc1 : 4d 4d 45 4c 44 55 4e 47   46
0bc9 : 92 20 20 20 20 20 20 20   3b
0bd1 : 20 20 20 20 20 20 20 20   d1
0bd9 : 20 20 20 20 20 20 20 20   d9
0be1 : 20 20 20 20 12 44 41 54   0e
0be9 : 45 4e 53 41 54 5a 2d 4e   bc
0bf1 : 52 2e 92 20 20 20 20 20   c7
0bf9 : 20 12 4e 41 4d 45 92 20   68
0c01 : 20 20 20 20 20 20 20 20   01
0c09 : 20 20 20 20 20 20 20 20   09
0c11 : 12 50 52 4f 47 52 41 4d   70
0c19 : 4d 2d 54 45 49 4c 92 20   3c
0c21 : 20 20 20 20 20 20 20 20   21
0c29 : 20 20 20 20 20 20 20 20   29
0c31 : 20 20 20 20 20 20 20 20   31
0c39 : 20 20 12 20 20 20 20 20   b6
0c41 : 20 20 20 20 20 20 20 20   41
0c49 : 20 20 20 20 20 20 20 20   49
0c51 : 20 20 20 20 20 20 20 20   51
0c59 : 20 20 20 20 20 20 20 20   59
0c61 : 20 20 20 92 0d 12 20 20   0e
0c69 : 20 20 20 20 20 20 20 20   69
0c71 : 20 20 20 20 20 20 20 20   71
0c79 : 20 20 20 20 20 20 20 20   79
0c81 : 20 20 20 20 20 20 20 20   81
0c89 : 20 20 20 20 20 20 92 00   13
0c91 : 20 45 49 4e 47 41 42 45   82
0c99 : 20 20 20 20 00 4c 4f 45   00
0ca1 : 53 43 48 45 4e 20 4d 49   fe
0ca9 : 54 20 3e 4a 3c 20 00 44   34
0cb1 : 41 54 45 49 20 45 52 4f   ab
0cb9 : 45 46 46 4e 45 4e 20 00   c4
0cc1 : 20 4c 4f 45 53 43 48 45   7f
0cc9 : 4e 20 20 20 20 20 20 20   f7
0cd1 : 00 20 53 50 45 49 43 48   fc
0cd9 : 45 52 4e 20 20 20 20 20   a3
0ce1 : 20 00 20 57 41 45 48 4c   ec
0ce9 : 45 4e 20 20 20 20 20 20   25
0cf1 : 20 20 00 44 49 53 4b 45   91
0cf9 : 54 54 45 4e 2d 49 4e 48   79
0d01 : 41 4c 54 00 46 41 4c 53   c4
0d09 : 43 48 45 52 20 44 41 54   de
0d11 : 45 49 4e 41 4d 45 00 20   f6
0d19 : 53 55 43 48 45 4e 20 20   78
0d21 : 20 20 20 20 20 00 20 46   6d
0d29 : 4f 52 4d 41 54 49 45 52   66
```

```
0d31 : 45 4e 20 20 20 20 00 20  ed
0d39 : 20 20 20 20 20 20 20 20  39
0d41 : 20 20 20 20 20 20 20 00  01
0d49 : 50 41 52 41 4d 45 54 45  d1
0d51 : 52 20 53 50 45 49 43 48  ce
0d59 : 45 52 4e 00 49 4e 44 45  fe
0d61 : 58 4b 41 52 54 45 49 20  ce
0d69 : 53 50 45 49 43 48 45 52  8f
0d71 : 4e 20 00 44 41 54 45 4e  c0
0d79 : 53 41 45 54 5a 45 20 46  26
0d81 : 52 45 49 47 45 42 45 4e  c9
0d89 : 20 00 50 41 52 41 4d 45  d5
0d91 : 54 45 52 20 4c 41 44 45  8b
0d99 : 4e 20 00 20 41 55 53 47  96
0da1 : 41 42 45 20 20 20 00 20  9c
0da9 : 41 45 4e 44 45 52 4e 20  09
0db1 : 20 20 00 46 31 3d 52 41  73
0db9 : 4e 44 20 20 46 33 3d 4d  c3
0dc1 : 49 54 54 45 20 20 46 35  79
0dc9 : 3d 4d 41 53 4b 45 20 20  07
0dd1 : 46 37 3d 54 45 58 54 00  f5
0dd9 : 20 52 45 54 54 45 4e 20  e7
0de1 : 20 20 20 00 20 41 45 4e  d7
0de9 : 44 45 52 4e 20 47 4c 4f  3a
0df1 : 42 41 4c 00 20 42 4c 41  af
0df9 : 45 54 54 45 52 4e 20 2b  95
0e01 : 2f 2d 20 00 20 44 49 53  bf
0e09 : 4b 20 2d 20 42 45 46 45  a6
0e11 : 48 4c 45 00 3d 3e 00 2d  f1
0e19 : 20 c2 49 54 54 45 20 d7  17
0e21 : 41 45 48 4c 45 4e 00 12  8b
0e29 : 20 46 37 20 92 20 20 20  29
0e31 : 47 45 4c 41 44 45 4e 45  88
0e39 : 20 44 41 54 45 49 20 41  f8
0e41 : 4b 54 49 56 49 45 52 45  66
0e49 : 4e 00 4c 51 45 cd 53 57  93
0e51 : 44 46 41 c4 43 cc c1 2b  99
0e59 : 2d 48 5f 5a 43 93 7a 90  a9
0e61 : 06 8a ce 97 a9 94 31 96  84
0e69 : ae 97 c7 92 45 96 6d 97  15
0e71 : e6 97 33 93 50 98 87 93  71
0e79 : b9 93 ff 9d a9 94 3d 99  17
0e81 : a9 00 85 62 a9 c0 85 63  55
0e89 : a9 e8 85 64 a9 04 85 65  30
0e91 : a9 e8 85 f3 a9 d8 85 f4  ef
0e99 : ae 5e 80 a0 00 b1 62 91  e5
0ea1 : 64 ad 3d 80 91 f3 c8 d0  b9
0ea9 : f4 e6 63 e6 65 e6 f4 ca  bd
0eb1 : d0 eb 60 a9 e8 85 62 a9  5c
0eb9 : 04 85 63 a9 00 85 64 a9  9f
0ec1 : c0 85 65 4c 65 86 a9 3d  d2
0ec9 : 85 62 a9 80 85 63 a9 00  14
0ed1 : 85 64 a9 c7 85 65 a0 d1  95
0ed9 : b1 62 91 64 88 c0 ff d0  dc
0ee1 : f7 60 a9 00 85 62 a9 c7  14
0ee9 : 85 63 a9 3d 85 64 a9 80  55
0ef1 : 85 65 4c ab 86 a9 00 8d  82
0ef9 : 55 80 8d 1c 80 8d 1d 80  5f
0f01 : 8d 1b 80 8d 4b 80 8d 62  a1
0f09 : 99 20 16 91 a2 01 20 29  6f
0f11 : 8b f0 f9 ad 00 cb 85 ac  16
0f19 : a2 12 dd 1e 86 f0 05 ca  99
0f21 : d0 f8 f0 0e 8a 0a aa e8  e1
0f29 : bd 2f 86 48 ca bd 2f 86  8d
0f31 : 48 60 20 11 87 a9 00 85  a4
0f39 : c6 4c ca 86 a9 0f a0 82  43
0f41 : 4c 8c 89 a2 ff a0 00 c8  21
0f49 : d0 fd 20 e4 ff d0 0b ca  05
0f51 : d0 f5 a5 c7 49 80 85 c7  bc
0f59 : 18 60 38 60 a2 04 20 ff  86
0f61 : e9 a2 04 a0 00 18 20 f0  d4
0f69 : ff 60 a2 06 20 ff e9 e8  7d
0f71 : e0 18 d0 f8 4c b4 87 20  79
0f79 : 59 87 a0 1a a9 20 20 d2  c3
0f81 : ff 88 d0 fa a2 00 a0 0e  20
0f89 : 18 20 f0 ff 60 20 6f 87  c1
0f91 : a0 10 a9 20 20 d2 ff 88  51
0f99 : d0 fa a2 01 a0 18 18 20  1b
0fa1 : f0 ff 60 a0 05 a9 00 99  8e
0fa9 : 00 cb 88 10 fa 20 8f 87  b1
0fb1 : a0 05 a9 20 20 d2 ff 88  ec
0fb9 : d0 fa a2 01 a0 0e 18 20  ea
0fc1 : f0 ff 60 a2 02 a0 0e 18  ab
0fc9 : 20 f0 ff 60 20 17 92 a9  c6
0fd1 : 04 8d 63 80 ad 20 80 d0  04
0fd9 : 03 4c 3f 87 20 55 86 20  ca
0fe1 : 42 91 20 31 87 a9 4f 85  28
0fe9 : d5 a9 50 a0 82 20 1e ab  b4
0ff1 : a2 18 20 ff e9 a2 06 a0  b4
0ff9 : 00 18 20 f0 ff a9 80 8d  96
1001 : 8a 02 a9 4f 85 d5 a6 d6  30
1009 : e0 18 10 e9 e0 06 30 e5  01
1011 : 20 3b 88 c9 93 f0 c2 c9  89
1019 : 91 f0 3a c9 9d f0 36 c9  b8
1021 : 11 f0 32 c9 1d f0 2e c9  16
1029 : 0d f0 2a c9 8c f0 34 c9  27
1031 : 88 f0 28 c9 85 d0 05 a9  bb
1039 : 3c 4c 2a 88 c9 86 d0 05  55
1041 : a9 3e 4c 2a 88 c9 87 d0  f8
1049 : 0c a9 12 20 d2 ff a9 3c  ff
1051 : 20 d2 ff a9 92 20 d2 ff  85
1059 : 4c d7 87 a9 1e 20 d2 ff  d6
1061 : 4c 5d 8f 4c d8 8f a9 20  ba
1069 : 85 02 a5 c6 85 cc 8d 92  4c
1071 : 02 f0 f3 78 a5 cf f0 0c  ac
1079 : a5 ce ae 87 02 a0 00 84  50
1081 : cf 20 13 ea 20 b4 e5 60  82
1089 : a9 00 8d 43 80 a0 3b 99  2b
1091 : 73 80 88 10 fa a0 02 99  58
1099 : 44 80 88 10 fa 8d 47 80  7c
10a1 : 8d e3 80 8d 4a 80 8d 48  61
10a9 : 80 8d 5b 80 8d 5c 80 8d  b0
10b1 : 6c 80 8d 4e 80 a9 05 8d  0f
10b9 : 5d 80 a9 00 a0 04 85 fc  fb
10c1 : 84 fd a2 04 a0 00 b1 fc  38
10c9 : c9 bc d0 06 ee 4e 80 8d  64
10d1 : 48 80 c9 3e d0 03 8d 48  7f
10d9 : 80 c9 3c d0 06 ee 4e 80  79
10e1 : 8d 48 80 ad 48 80 f0 05  be
10e9 : 20 04 89 b0 41 ee 5c 80  82
10f1 : ad 5c 80 c9 28 d0 08 ee  2d
10f9 : 5b 80 a9 00 8d 5c 80 4e  59
1101 : 4e 80 90 1e 8c 42 80 ac  ad
1109 : 6c 80 c0 1e f0 11 ad 5b  ae
1111 : 80 99 91 80 c8 ad 5c 80  3f
1119 : 99 91 80 c8 8c 6c 80 ac  3b
1121 : 42 80 c8 d0 a1 e6 fd ca  ce
1129 : d0 9c ee e3 80 18 60 10  ea
1131 : 21 49 80 8d 48 80 98 ac  0d
1139 : 4a 80 c0 03 d0 03 4c 88  bb
1141 : 89 48 ad 43 80 99 44 80  a9
1149 : ee 4a 80 ee 47 80 68 a8  c6
1151 : 18 60 c9 3c d0 0d ee 47  53
1159 : 80 ad 47 80 c9 fe d0 f0  4b
1161 : 4c 73 89 ad 43 80 c9 0f  fc
1169 : d0 03 4c 7a 89 8c 60 80  9d
1171 : ce 47 80 ac 43 80 ad e3  4f
1179 : 80 99 82 80 18 6d 47 80  82
1181 : 90 03 4c 81 89 8d e3 80  6c
1189 : ad 47 80 99 73 80 ee 43  ab
1191 : 80 ac 60 80 a9 00 8d 47  ef
1199 : 80 8d 48 80 f0 b2 a9 0e  69
11a1 : a0 81 4c 8c 89 a9 24 a0  5e
11a9 : 81 4c 8c 89 a9 3a a0 81  97
11b1 : 4c 8c 89 a9 50 a0 81 85  f6
11b9 : fd 84 fe 20 4f 9f ad 3e  e1
11c1 : 80 8d 86 02 20 59 87 a5  20
11c9 : fd a4 fe 20 1e ab 20 18  cc
11d1 : 87 90 f1 a9 00 85 c7 20  dd
11d9 : 4c 87 38 60 ad 20 80 d0  82
11e1 : 28 20 17 92 20 ff 8d b0  cb
11e9 : 46 20 a9 91 20 2c 8e b0  db
11f1 : 3e ad 4a 80 f0 05 20 a0  a1
11f9 : 96 b0 31 a9 00 8d 56 80  30
1201 : 8d 1f 80 8d 13 80 20 aa  fb
1209 : 9e 20 aa 8c 20 eb 8b b0  e4
1211 : 1b 20 fd 8b b0 16 18 20  89
1219 : 2c 94 20 17 92 ad 59 80  77
1221 : 8d 20 d0 ad 5a 80 8d 21  ca
1229 : d0 4c 07 8a 20 0d 8f 4c  74
1231 : d8 8f a9 00 85 c7 20 27  a1
1239 : 91 20 55 86 20 6f 8d 90  d5
1241 : 03 4c ca 86 20 31 91 20  00
1249 : cb 8d 20 31 87 a9 00 8d  ea
1251 : 60 80 8d 6c 80 8d 49 80  7d
1259 : 8d 42 80 ad 43 80 8d 47  da
1261 : 80 20 ff 8a ae 49 80 bd  f5
1269 : 73 80 aa 20 29 8b d0 24  45
1271 : ad 62 80 c9 85 d0 1d ad  57
1279 : 49 80 48 ad 47 80 48 a9  b7
1281 : 01 8d 47 80 20 40 8d 20  a5
1289 : b9 8b 68 8d 47 80 68 8d  09
1291 : 49 80 10 0a ad 60 80 18  70
1299 : 6d 41 80 8d 60 80 ee 49  d1
12a1 : 80 ce 47 80 d0 bb ad 1d  46
12a9 : 80 d0 16 ae 42 80 a9 0d  d6
12b1 : 9d 00 ce 20 99 8a ad 4b  41
12b9 : 80 d0 03 4c 07 8a 4c ca  77
12c1 : 86 4c d9 96 20 bd 8a 20  11
12c9 : 23 8c b0 14 20 3b 8c b0  50
12d1 : 0f a9 00 8d 4e 80 20 09  e2
12d9 : 8d 20 5d 94 20 e8 95 18  30
12e1 : 60 a9 85 a0 82 4c 8c 89  5b
12e9 : a9 00 8d 6c 80 8d 42 80  02
12f1 : ad 43 80 8d 47 80 20 ff  0b
12f9 : 8a 20 14 8b c9 3e f0 1c  94
1301 : 48 a9 4f 85 d5 a5 9a c9  2b
1309 : 03 f0 02 e6 d3 68 20 d2  88
1311 : ff ac 42 80 99 00 ce ee  ba
1319 : 42 80 d0 dd ce 47 80 d0  56
1321 : d5 ac 42 80 a9 0d 99 00  56
1329 : ce 60 ac 6c 80 b9 91 80  fd
1331 : aa c8 b9 91 80 c8 8c 6c  39
1339 : 80 a8 18 20 f0 ff 60 a4  f1
1341 : d3 b1 d1 85 d7 29 3f 06  e2
1349 : d7 24 d7 10 02 09 80 70  76
1351 : 02 09 40 60 ad 3e 80 8d  de
1359 : 86 02 a9 20 85 02 8e 41  74
1361 : 80 8a 18 65 d3 85 d5 a2  df
1369 : 00 8e 61 80 20 3b 88 ae  74
1371 : 61 80 8d 62 80 c9 02 f0  02
1379 : 78 c9 0d f0 67 c9 85 f0  f4
1381 : 63 c9 14 f0 26 c9 9d f0  f5
1389 : 4f ec 41 80 f0 db ad 62  18
1391 : 80 c9 94 f0 26 c9 1d f0  40
1399 : 0b 20 d4 8b b0 ce ad 62  58
13a1 : 80 9d 00 cb 20 d2 ff e8  d4
13a9 : 4c 3e 8b 8a f0 bb 20 5f  75
13b1 : 99 4c 3e 8b a9 20 91 d1  f7
13b9 : 4c 3e 8b a4 d5 20 24 ea  60
13c1 : 88 b1 d1 c8 c4 d5 f0 08  7e
13c9 : 91 d1 88 b1 f3 c8 91 f3  4f
13d1 : 88 c4 d3 d0 eb 4c 89 8b  29
13d9 : 8a f0 8e a9 9d 20 d2 ff  da
13e1 : ca 4c 3e 8b ae 3d 80 8e  c6
13e9 : 86 02 ae 41 80 ac 61 80  38
13f1 : 60 20 cc ff a0 00 8c 55  7b
13f9 : 80 a2 fd 9a 4c ca 86 10  f2
1401 : 03 4c e3 8b c9 20 90 0c  8d
1409 : c9 60 b0 08 18 60 29 7f  58
1411 : c9 20 b0 f8 38 60 a9 0f  81
1419 : a2 08 a0 0f 20 ba ff a9  f4
1421 : 00 20 bd ff 20 c0 ff 60  69
1429 : a9 02 a2 08 a0 05 20 ba  a5
1431 : ff ac db 80 a2 00 bd e0  70
1439 : 80 99 c6 80 c8 e8 e0 04  a7
1441 : d0 f4 98 a2 c6 a0 80 20  b9
1449 : bd ff 20 c0 ff 60 20 cc  43
1451 : ff a2 0f 20 c9 ff a2 00  90
1459 : bd 6d 80 20 d2 ff e8 e0  83
1461 : 06 d0 f5 4c 19 8f 20 cc  fe
1469 : ff a2 02 20 c9 ff b0 20  de
1471 : a2 00 bd 00 ce 20 d2 ff  bc
1479 : e8 ec e3 80 d0 f4 20 cc  af
1481 : ff 20 19 8f b0 09 a9 01  c5
1489 : 8d 56 80 8d 1e 80 18 60  1a
1491 : 4c 19 8f ad 1e 80 f0 09  bf
1499 : 20 aa 8c 20 eb 8b 20 fd  cd
14a1 : 8b 20 23 8c b0 2c 20 cc  1d
14a9 : ff a2 02 20 c6 ff b0 e0  6f
14b1 : a2 00 8e 1e 80 20 cf ff  03
14b9 : 9d 00 ce e8 ec e3 80 d0  b8
14c1 : f4 20 cc ff ad 00 ce 8d  2a
14c9 : 4e 80 a5 90 c9 40 d0 02  b9
14d1 : 18 60 4c 19 8f 20 cc ff  7d
14d9 : ad 69 80 20 c3 ff a9 02  46
14e1 : 20 c3 ff ad 66 80 20 c3  0b
14e9 : ff a9 0f 20 c3 ff 60 20  83
14f1 : cc ff 20 9f 91 a2 02 20  2f
14f9 : c9 ff a9 ff 20 d2 ff 4c  5d
1501 : 19 8f 60 a0 00 a2 c3 a9  85
1509 : 00 8d 00 c3 84 60 86 61  70
1511 : a0 01 ae 63 80 a9 ff 91  c2
1519 : 60 c8 d0 fb e6 61 ca d0  d7
1521 : f6 a0 01 8c 70 80 8c 65  41
1529 : 80 88 8c 6f 80 8c 64 80  fd
1531 : 8c 20 80 60 a9 00 18 6d  cf
1539 : 6f 80 8d 1e 8d a9 c3 6d  1f
1541 : 70 80 8d 1f 8d ad 4e 80  b9
1549 : 8d 00 00 60 ad 4e 80 c9  c5
1551 : ff f0 46 a9 4f 85 d5 a9  5b
1559 : 00 8d 6c 80 8d 49 80 8d  8b
1561 : 60 80 ad 43 80 8d 47 80  68
1569 : 20 ff 8a ae 49 80 bd 73  78
1571 : 80 aa ad 3e 80 8d 86 02  0c
1579 : ac 60 80 b9 00 ce 85 d7  e9
1581 : 20 d4 8b b0 14 a5 d7 20  12
1589 : d2 ff c8 ca d0 ed 8c 60  56
1591 : 80 ee 49 80 ce 47 80 d0  b5
1599 : cf 60 a0 00 a2 c3 84 60  dc
15a1 : 86 61 a0 00 ae 65 80 b1  7b
15a9 : 60 d0 3e c8 d0 f9 e6 61  55
15b1 : ca d0 f4 ac 65 80 cc 63  0a
15b9 : 80 f0 25 c8 8c 65 80 8c  23
15c1 : 70 80 a0 00 8c 6f 80 20  20
15c9 : 23 8c b0 05 20 c4 8c 90  7b
15d1 : c9 ce 65 80 ce 70 80 ad  38
15d9 : 65 80 8d 63 80 4c 6f 8d  91
15e1 : 20 30 92 20 31 87 4c b6  b0
15e9 : 8a 8c 6f 80 a5 61 38 e9  bf
15f1 : c3 8d 70 80 18 60 ad 3e  5f
15f9 : 80 8d 86 02 20 82 87 ac  af
1601 : 6f 80 ad 70 80 20 95 b3  f0
```

Listing 1. Das universelle Dateiverwaltungsprogramm »Datev«. Beachten Sie bitte die Eingabehinweise auf Seite 100.

```
1609 : 20 dd bd a2 00 bd 00 01  cb
1611 : f0 03 e8 d0 f8 e0 05 f0  63
1619 : 08 a9 20 9d 00 01 e8 d0  ff
1621 : f4 a9 00 9d 00 01 20 1e  62
1629 : ab 60 20 31 87 20 62 87  45
1631 : a2 10 20 29 8b f0 1e 8c  da
1639 : da 80 98 a2 00 a0 cb 20  42
1641 : bd ff ad 66 80 ae 67 80  52
1649 : ac 68 80 20 ba ff a9 00  a0
1651 : 20 90 ff 18 60 38 60 20  46
1659 : c0 ff b0 40 ae 66 80 20  ad
1661 : c6 ff b0 38 20 cf ff 8d  f6
1669 : 00 c0 a5 90 d0 2e 20 cf  e3
1671 : ff 8d 01 c0 c9 c0 d0 27  c3
1679 : 20 cc ff a9 00 a8 a2 c0  86
1681 : 20 d5 ff b0 17 a0 06 a9  84
1689 : a0 d9 00 c0 d0 11 88 10  06
1691 : f8 20 b7 86 a9 01 8d 20  71
1699 : 80 4c e3 8e 4c 0d 8f 20  b6
16a1 : e3 8e a9 d9 a0 84 4c 8c  e9
16a9 : 89 ac da 80 a9 00 99 b1  b3
16b1 : 80 88 b9 00 cb 99 b1 80  35
16b9 : 88 10 f7 ac da 80 c8 c8  43
16c1 : 8c c5 80 ac da 80 c0 08  aa
16c9 : 90 02 a0 08 8c 42 80 b9  d4
16d1 : b1 80 99 c6 80 88 d0 f7  81
16d9 : a9 5f 99 c6 80 ac 42 80  e8
16e1 : a2 00 bd dc 80 99 c6 80  7f
16e9 : c8 e8 e0 04 d0 f4 8c db  7d
16f1 : 80 60 20 9b 86 20 88 86  b5
16f9 : a9 00 85 60 a9 c0 85 61  89
1701 : a2 00 a0 c9 a9 60 20 d8  d4
1709 : ff 20 b7 ff d0 33 20 cc  c7
1711 : ff ad 66 80 20 c3 ff 18  e1
1719 : 60 a9 00 85 60 a9 20 85  dd
1721 : 61 a2 00 ac 4a 80 88 b9  a7
1729 : 3a 80 a8 a9 60 20 d8 ff  6d
1731 : 20 b7 ff d0 0c 4c e3 8e  17
1739 : 20 aa 8c a5 90 c9 80 f0  c1
1741 : 37 20 eb 8b 20 cc ff a2  a2
1749 : 0f 20 c6 ff b0 ea a0 00  ff
1751 : 20 cf ff 99 00 cf c8 24  76
1759 : 90 50 f5 88 a9 00 85 90  72
1761 : 99 00 cf 20 cc ff ad 00  76
1769 : cf c9 30 f0 12 c9 35 f0  6d
1771 : 0e a9 03 a0 cf 4c 8c 89  cd
1779 : a9 f7 a0 81 4c 8c 89 ad  21
1781 : 01 cf c9 30 d0 eb 18 60  70
1789 : a9 00 8d 8a 02 20 31 87  dc
1791 : 20 5d 88 90 03 4c b7 87  14
1799 : 20 ff 8d b0 f8 20 7e 8e  da
17a1 : 20 8b 91 20 01 91 a9 00  32
17a9 : 8d 4e 80 20 d8 8c 20 af  53
17b1 : 8f 20 21 93 a9 00 8d 57  8b
17b9 : 80 20 c7 8e b0 d7 ad 4a  22
17c1 : 80 f0 05 20 bf 90 b0 cd  de
17c9 : 20 eb 8b 20 fd 8b 20 23  c9
17d1 : 8c 20 c4 8c 20 0d 8f 4c  71
17d9 : d8 8f a9 00 85 60 a0 20  01
17e1 : 84 61 a2 60 a8 a9 00 91  c5
17e9 : 60 c8 d0 fb e6 61 ca d0  a7
17f1 : f6 60 a9 0e 20 d2 ff ae  39
17f9 : 59 80 8e 20 d0 ae 5a 80  27
1801 : 8e 21 d0 20 44 e5 20 aa  a1
1809 : 8c 78 a9 31 a0 ea 8d 14  22
1811 : 03 8c 15 03 58 a9 20 85  5e
1819 : 02 ae 3d 80 8e 86 02 a9  4a
1821 : be a0 82 20 1e ab 20 28  e4
1829 : 9f ad 20 80 f0 0f a2 08  d9
1831 : a0 00 18 20 f0 ff a9 fc  8b
1839 : a0 85 20 1e ab 20 01 91  4b
1841 : a2 10 a0 01 18 20 f0 ff  7a
1849 : a9 ec a0 85 20 1e ab a2  28
1851 : 10 a0 01 18 20 f0 ff 20  be
1859 : aa 9e 20 3b 88 c9 85 d0  50
1861 : 08 a9 00 8d 20 80 4c b1  8a
1869 : 89 c9 86 d0 03 4c 6f 90  04
1871 : c9 87 d0 03 4c a1 87 c9  16
1879 : 03 d0 03 4c fa 90 c9 89  9d
1881 : d0 09 20 17 92 20 12 99  66
1889 : 4c d8 8f ae 20 80 f0 ca  5a
1891 : c9 88 d0 c6 20 b7 86 4c  1e
1899 : b1 89 20 17 92 20 5e 91  c1
18a1 : 20 56 99 4c d8 8f 20 31  c9
18a9 : 87 20 50 91 20 aa 8c 6e  ed
18b1 : 56 80 90 06 20 b7 90 4c  c7
18b9 : d8 8f 6e 13 80 90 03 20  30
18c1 : 9a 90 4c d8 8f 20 db 90  5c
18c9 : 20 55 86 20 9b 86 ad da  94
18d1 : 80 a2 b1 a0 80 20 14 8e  99
18d9 : 20 c0 ff 20 8b 91 20 c7  b3
18e1 : 8e 60 20 9a 90 ad 4a 80  9b
18e9 : f0 d8 a9 5e 8d b1 80 20  24
18f1 : db 90 ad da 80 a2 b1 a0  00
18f9 : 80 20 14 8e 20 c0 ff 20  a8
1901 : 95 91 20 ee 8e 60 ad c5  73
1909 : 80 a2 af a0 80 20 bd ff  da
1911 : a9 0f a2 08 a0 0f 20 ba  64
1919 : ff 20 c0 ff 20 cc ff a9  14
1921 : 0f 20 c3 ff 60 a9 00 85  8f
1929 : 02 4c e2 fc a2 18 a0 00  17
1931 : 18 20 f0 ff ad 3d 80 8d  77
1939 : 86 02 a9 92 a0 82 4c 0f  ea
1941 : 92 a9 0c a0 85 20 00 92  3d
1949 : 20 31 87 a9 60 a0 83 4c  cb
1951 : 09 92 20 62 87 a9 b1 a0  c5
1959 : 80 4c 09 92 a9 65 a0 84  e5
1961 : 4c 00 92 20 31 87 a9 72  31
1969 : a0 84 4c 09 92 a9 84 a0  49
1971 : 84 4c 00 92 a9 95 a0 84  41
1979 : 4c 00 92 a9 a6 a0 84 4c  b9
1981 : 00 92 a9 ec a0 84 4c 00  32
1989 : 92 a9 c8 a0 84 4c 00 92  06
1991 : a9 b7 a0 84 4c 00 92 a9  31
1999 : 70 a0 85 4c 00 92 a9 7c  78
19a1 : a0 85 4c 00 92 a9 fb a0  bf
19a9 : 84 4c 00 92 20 31 87 a9  a3
19b1 : 2a a0 82 4c 09 92 20 31  5e
19b9 : 87 a9 1d a0 85 4c 09 92  74
19c1 : 20 31 87 a9 31 a0 85 4c  58
19c9 : 09 92 20 31 87 a9 48 a0  72
19d1 : 85 4c 09 92 20 31 87 a9  0e
19d9 : 5f a0 85 4c 09 92 20 31  7b
19e1 : 87 a9 6c a0 83 4c 09 92  50
19e9 : 20 31 87 a9 e6 a0 80 4c  c7
19f1 : 09 92 20 31 87 a9 88 a0  9b
19f9 : 85 4c 09 92 20 31 87 a9  36
1a01 : ad a0 85 4c 00 92 20 31  61
1a09 : 87 a9 c9 a0 85 4c 00 92  cb
1a11 : 20 31 87 a9 d9 a0 85 4c  32
1a19 : 00 92 20 31 87 a9 e9 a0  3f
1a21 : 85 4c 09 92 20 31 87 a9  5e
1a29 : b9 a0 85 48 98 48 20 98  1a
1a31 : 87 68 a8 68 ae 3e 80 8e  1f
1a39 : 86 02 a2 4f 86 d5 20 1e  27
1a41 : ab 60 20 44 e5 a9 00 8d  74
1a49 : 15 d0 ad 3d 80 8d 86 02  6c
1a51 : a9 8f a0 83 20 1e ab 20  3c
1a59 : 01 91 60 20 cb 8d 20 68  b9
1a61 : 8c 20 55 86 20 21 8d ad  c0
1a69 : 55 80 d0 01 60 ad 69 80  6d
1a71 : 20 c3 ff ad 69 80 85 b8  4a
1a79 : ad 6b 80 85 b9 ad 6a 80  60
1a81 : 85 ba 20 c0 ff ae 69 80  a0
1a89 : 20 c9 ff a0 e6 a2 80 84  30
1a91 : 7a 86 7b 20 79 00 f0 4c  25
1a99 : b0 28 20 f3 bc 20 9b bc  99
1aa1 : c6 65 a5 65 cd 43 80 90  4a
1aa9 : 02 b0 e8 8d 49 80 0a 8d  cb
1ab1 : 6c 80 a9 01 8d 47 80 a9  50
1ab9 : 00 8d 42 80 20 cb 8a 4c  44
1ac1 : 68 92 c9 2c d0 0b a9 0d  90
1ac9 : 20 d2 ff 20 73 00 4c 68  8f
1ad1 : 92 c9 3b d0 05 a9 20 4c  e7
1ad9 : 9d 92 c9 2a d0 03 20 cb  b4
1ae1 : 8d 4c a0 92 20 cc ff 18  a7
1ae9 : 60 a9 cb a0 81 20 8c 89  83
1af1 : 4c ca 86 20 7a 91 20 bd  78
1af9 : 91 20 36 87 a2 26 20 29  47
1b01 : 8b f0 46 20 36 87 a0 00  bc
1b09 : 8c 53 80 ac 53 80 a2 0f  d6
1b11 : a4 d3 b1 d1 dd 29 80 f0  50
1b19 : 0d ca d0 f8 49 80 91 d1  61
1b21 : 20 11 87 4c c8 92 ee 53  b9
1b29 : 80 e6 d3 ac 53 80 cc 61  d6
1b31 : 80 d0 d8 20 21 93 20 36  ef
1b39 : 87 a2 00 a4 d3 b1 d1 9d  f3
1b41 : e6 80 e6 d3 e8 e0 26 d0  6b
1b49 : f2 4c ca 86 a0 25 a9 20  ff
1b51 : 99 e6 80 88 10 fa a9 01  10
1b59 : 8d 57 80 8d 13 80 60 ad  76
1b61 : 1f 80 d0 03 4c c5 97 a9  f9
1b69 : 01 8d 1c 80 4c 49 93 a9  f9
1b71 : 00 8d 1c 80 20 49 91 20  22
1b79 : 38 91 a2 01 20 29 8b f0  9e
1b81 : 2f ad 00 cb c9 4a d0 28  83
1b89 : ad 1c 80 f0 03 4c d9 96  aa
1b91 : 20 55 86 20 bd 8a 20 23  f8
1b99 : 8c 20 3b 8c a9 ff 8d 4e  03
1ba1 : 80 20 09 8d 20 c8 96 20  08
1ba9 : e8 95 ad 1c 80 f0 01 60  9f
1bb1 : 4c ca 86 20 db 91 ad 70  ea
1bb9 : 80 cd 65 80 f0 0a ee 6f  83
1bc1 : 80 d0 0f ee 70 80 10 0a  aa
1bc9 : a2 00 a0 01 8e 70 80 8c  3b
1bd1 : 6f 80 20 30 92 20 e4 ff  4c
1bd9 : f0 fb c9 2b f0 d5 c9 2d  de
1be1 : f0 03 4c ca 86 20 db 91  bb
1be9 : ac 6f 80 ad 70 80 d0 04  79
1bf1 : c0 01 f0 12 88 8c 6f 80  5c
1bf9 : c0 ff d0 d6 ad 70 80 f0  0a
1c01 : 05 ce 70 80 10 cc ae 65  86
1c09 : 80 88 f0 c0 20 31 87 20  0b
1c11 : 78 87 a2 04 20 29 8b d0  91
1c19 : 04 38 4c 21 94 a9 00 a0  48
1c21 : cb 85 7a 84 7b 20 79 00  7d
1c29 : 20 f3 bc 20 9b bc a5 64  75
1c31 : cd 65 80 b0 18 90 04 a5  48
1c39 : 65 d0 12 a5 64 d0 04 c5  a8
1c41 : 65 f0 11 8d 70 80 a5 65  81
1c49 : 8d 6f 80 18 60 a9 66 a0  df
1c51 : 81 4c 8c 89 4c 11 87 78  a9
1c59 : a9 39 a0 94 8d 14 03 8c  f8
1c61 : 15 03 58 60 ad 8d 02 cd  05
1c69 : 21 80 f0 07 8d 21 80 c9  5f
1c71 : 02 f0 03 4c 31 ea a6 c6  c8
1c79 : a9 02 9d 77 02 e8 86 c6  89
1c81 : a9 7f 8d 00 dc 4c 81 ea  59
1c89 : ae 4a 80 f0 47 a0 1b a9  d4
1c91 : 20 99 00 cb 88 10 fa 20  2d
1c99 : c8 96 a0 ff c8 cc 4a 80  f1
1ca1 : f0 32 8c 60 80 b9 4f 80  ee
1ca9 : 48 b9 44 80 a8 b9 73 80  16
1cb1 : 8d 42 80 ce 42 80 b9 82  6d
1cb9 : 80 aa 68 a8 bd 00 ce c9  68
1cc1 : 20 d0 02 a9 0d 99 00 cb  34
1cc9 : c8 e8 ce 42 80 10 ed ac  9b
1cd1 : 60 80 10 c8 60 ae 4a 80  34
1cd9 : f0 4b 20 31 87 20 55 86  79
1ce1 : 20 57 91 a0 ff c8 cc 4a  33
1ce9 : 80 f0 26 8c 60 80 b9 44  76
1cf1 : 80 8d 49 80 0a 8d 6c 80  5a
1cf9 : a8 20 ff 8a ae 49 80 bd  b5
1d01 : 73 80 c9 09 90 02 a9 08  18
1d09 : aa 20 29 8b ac 60 80 10  6f
1d11 : d4 20 bd 8a 20 5d 94 a5  41
1d19 : ac c9 5f d0 03 4c b4 98  32
1d21 : 20 17 95 90 03 4c ca 86  0f
1d29 : 20 81 91 a2 01 20 29 8b  8f
1d31 : ad 00 cb 29 7f c9 44 f0  30
1d39 : 04 c9 4d d0 eb 09 80 4c  31
1d41 : eb 86 a9 00 8d 1f 80 20  ee
1d49 : 57 91 a0 7f a9 00 99 00  82
1d51 : cc 88 10 fa c8 84 61 84  04
1d59 : 62 a0 00 8c 47 80 ad 4d  67
1d61 : 80 8d 42 80 ad 42 80 85  42
1d69 : 64 a5 62 85 63 a5 61 0a  e6
1d71 : 26 63 0a 26 63 0a 26 63  76
1d79 : 48 a5 63 05 64 85 64 68  e2
1d81 : 85 63 ac 47 80 b9 4f 80  e0
1d89 : aa a0 00 b1 63 f0 33 bd  c0
1d91 : 00 cb c9 0d f0 12 09 80  50
1d99 : d1 63 f0 06 29 7f d1 63  b5
1da1 : d0 20 e8 c8 c0 08 d0 e3  2c
1da9 : ee 47 80 ad 47 80 cd 4a  55
1db1 : 80 f0 0c ad 42 80 18 69  bd
1db9 : 20 8d 42 80 4c 39 95 20  66
1dc1 : b1 95 e6 61 d0 93 e6 62  2d
1dc9 : a5 62 c9 04 d0 8b ad 1f  f1
1dd1 : 80 f0 02 18 60 a9 7e a0  dc
1dd9 : 81 4c 8c 89 c6 61 a5 61  a5
1de1 : 8d e4 80 a5 62 8d e5 80  e0
1de9 : 6e e5 80 6e e4 80 6e e5  10
1df1 : 80 6e e4 80 6e e5 80 6e  e6
1df9 : e4 80 ac e4 80 a5 61 e6  6d
1e01 : 61 29 07 aa b9 00 cc 1d  17
1e09 : 22 80 99 00 cc a9 ff 8d  07
1e11 : 1f 80 60 ae 4a 80 f0 44  53
1e19 : 8e 42 80 a2 00 8e 47 80  cf
1e21 : ad 6f 80 ae 70 80 86 62  66
1e29 : 0a 26 62 0a 26 62 0a 26  0a
1e31 : 62 85 61 ad 4d 80 05 62  16
1e39 : 85 62 ee 47 80 ad 47 80  27
1e41 : 0a 0a 0a aa ca a0 07 bd  71
1e49 : 00 cb 91 61 ca 88 10 f7  e0
1e51 : a5 62 18 69 20 85 62 ce  b0
1e59 : 42 80 d0 de 60 20 65 91  ab
1e61 : 20 e1 93 90 06 20 cb 8d  14
1e69 : 4c ca 86 20 30 92 4c ca  1e
1e71 : 86 20 73 91 a9 01 8d 4b  86
1e79 : 80 4c 1c 8a 20 d1 91 a0  90
1e81 : 00 a2 c3 84 fb 86 fc a0  7d
1e89 : 01 ae 63 80 b1 fb d0 32  6d
1e91 : 8c 53 80 8e 54 80 8c 6f  13
1e99 : 80 a5 fc 38 e9 c3 8d 70  06
1ea1 : 80 20 30 92 ad 55 80 d0  b9
1ea9 : 03 20 99 8a 20 e1 ff d0  27
1eb1 : 0b 20 b3 91 a2 00 20 29  e8
1eb9 : 8b 20 31 87 ac 53 80 ae  56
1ec1 : 54 80 c8 d0 c7 e6 fc ca  df
1ec9 : d0 c2 60 a0 00 a9 5e 99  20
1ed1 : 00 cb c8 cc da 80 f0 0c  10
1ed9 : b9 b1 80 99 00 cb c8 cc  d9
1ee1 : da 80 d0 f4 20 0f 8e 20  c3
1ee9 : c0 ff a9 00 a8 a2 20 20  74
1ef1 : d5 ff 60 a9 00 a0 02 be  9e
1ef9 : 4f 80 9d 00 cb 88 10 f7  21
1f01 : 60 20 6c 91 a0 ff 8c 53  a1
```

```
1f09 : 80 ac 53 80 c8 8c 53 80   03
1f11 : 30 7c b9 00 cc f0 f2 8d   29
1f19 : 52 80 a9 00 8d 70 80 98   a5
1f21 : 0a 2e 70 80 0a 2e 70 80   43
1f29 : 0a 2e 70 80 8d e4 80 a2   be
1f31 : ff e8 e0 08 f0 d3 ad 52   e6
1f39 : 80 3d 22 80 f0 f3 8e 54   82
1f41 : 80 e8 8a 18 6d e4 80 8d   f6
1f49 : 6f 80 20 cb 8d ad 1c 80   32
1f51 : f0 06 20 65 93 4c 5e 97   3d
1f59 : ad 1d 80 f0 03 4c 67 98   34
1f61 : 20 33 92 b0 31 ad 55 80   ac
1f69 : f0 15 20 e1 ff d0 1a 20   57
1f71 : b3 91 a2 00 86 c6 20 29   07
1f79 : 8b 20 31 87 4c 5e 97 20   a8
1f81 : b3 91 a2 00 86 c6 20 29   17
1f89 : 8b ae 54 80 10 a3 ad 1b   9c
1f91 : 80 f0 03 4c e5 98 4c ca   be
1f99 : 86 20 31 87 20 59 99 90   c1
1fa1 : 03 4c be 92 ad 4a 80 f0   dd
1fa9 : 12 ad 1f 80 f0 42 ad 57   f0
1fb1 : 80 f0 1b a9 01 8d 55 80   78
1fb9 : 4c d6 96 ad 57 80 f0 0e   25
1fc1 : a9 01 8d 55 80 20 6c 91   d7
1fc9 : 20 54 96 4c ca 86 a9 b4   33
1fd1 : a0 81 20 8c 89 a9 46 4c   63
1fd9 : eb 86 20 31 87 20 59 99   48
1fe1 : 90 03 4c be 92 ad 57 80   d3
1fe9 : f0 e4 20 42 92 4c ca 86   5f
1ff1 : a9 e2 a0 81 20 8c 89 4c   89
1ff9 : ca 86 20 31 87 ad 4a 80   45
2001 : d0 03 4c ca 86 ad 1f 80   13
2009 : f0 e6 a9 00 8d 55 80 4c   f5
2011 : d6 96 20 c7 91 20 e4 ff   e1
2019 : f0 fb c9 0d f0 5b c9 85   37
2021 : d0 0c ee 20 d0 ac 20 d0   4c
2029 : 8c 59 80 4c 46 98 c9 86   69
2031 : d0 0c ee 21 d0 ac 21 d0   80
2039 : 8c 5a 80 4c 46 98 c9 87   fb
2041 : d0 14 ee 3d 80 ad 3d 80   ea
2049 : 29 0f 8d 3d 80 20 17 92   8f
2051 : 20 55 86 4c 3d 98 c9 88   18
2059 : d0 b8 ee 3e 80 ad 3e 80   78
2061 : 29 0f 8d 3e 80 8d 86 02   d0
2069 : 20 cb 8d 20 27 91 20 21   98
2071 : 8d a9 01 8d 13 80 4c e7   fb
2079 : 97 4c ca 86 ad 1f 80 f0   71
2081 : 5b a9 01 8d 1d 80 20 78   ea
2089 : 87 20 f9 91 20 55 86 4c   30
2091 : 1f 8a 20 68 8c a0 00 8c   f1
2099 : 6c 80 8c 42 80 8c 60 80   a0
20a1 : ad 43 80 8d 47 80 20 ff   bb
20a9 : 8a 20 14 8b c9 20 f0 1e   57
20b1 : ac 60 80 b9 82 80 8d 42   cb
20b9 : 80 20 14 8b c9 3e f0 0e   2e
20c1 : ac 42 80 99 00 ce 20 d2   7e
20c9 : ff ee 42 80 d0 eb ee 60   c9
20d1 : 80 ce 47 80 d0 d0 20 9c   e7
20d9 : 8a 4c 5e 97 4c c5 97 a0   a6
20e1 : ff c8 cc 4a 80 d0 03 4c   f4
20e9 : f5 94 8c 60 80 b9 4f 80   6c
20f1 : a8 b9 00 cb c9 5f f0 05   55
20f9 : ac 60 80 10 e4 8c 18 80   0c
2101 : 20 31 87 20 59 99 90 03   4a
2109 : 4c be 92 a9 1f 8d 14 80   3e
2111 : ee 14 80 ad 14 80 c9 5f   0a
2119 : f0 20 ac 18 80 99 00 cb   b4
2121 : 20 e1 ff f0 15 a9 01 8d   0d
2129 : 1f 80 8d 1b 80 20 1c 95   f4
2131 : ad 1f 80 c9 01 f0 d9 4c   5f
2139 : 79 97 4c ca 86 20 e5 91   0e
2141 : 20 ef 91 a2 23 20 29 8b   00
2149 : f0 1c 20 eb 8b 20 cc ff   ba
2151 : a2 0f 20 c9 ff b0 0f ac   d7
2159 : 61 80 a2 00 bd 00 cb 20   ee
2161 : d2 ff e8 88 d0 f6 4c 0d   8e
2169 : 8f ad 00 08 f0 10 ad 18   46
2171 : d0 c9 17 f0 04 a9 17 d0   95
2179 : 02 a9 13 8d 18 d0 4c ca   95
2181 : 86 4c 7a 99 4c 2d 9a 4c   30
2189 : 85 9c 4c 35 9d 00 57 45   d8
2191 : 49 54 45 52 20 3e 3e d2   33
2199 : c5 d4 d5 d2 ce 3c 3c 00   58
21a1 : 24 00 00 00 50 a9 01 a2   61
21a9 : 75 a0 99 20 bd ff a9 02   5f
21b1 : a8 a2 08 20 ba ff a9 60   c3
21b9 : 85 b9 20 d5 f3 a5 ba 20   75
21c1 : b4 ff a5 b9 20 96 ff a9   1f
21c9 : 00 85 90 20 09 9a a0 03   a2
21d1 : 84 b7 20 a5 ff 85 c3 20   69
21d9 : a5 ff 85 c4 a4 90 d0 38   fa
21e1 : a4 b7 88 d0 eb a6 c3 a5   eb
21e9 : c4 20 cd bd a9 20 20 16   31
21f1 : e7 20 a5 ff a6 90 d0 20   c4
21f9 : c9 00 f0 0a 20 16 e7 20   d2
2201 : e1 ff f0 14 d0 eb ce 76   35
2209 : 99 d0 03 20 f0 99 a9 0d   6c
2211 : 20 16 e7 a0 02 4c a5 99   97
2219 : 20 42 f6 a2 04 20 ff e9   81
2221 : a2 04 a0 00 18 20 f0 ff   34
2229 : a9 63 a0 99 20 1e ab 20   c1
2231 : e4 ff f0 fb 20 16 9a a2   33
2239 : 07 20 ff e9 e8 e0 17 d0   21
2241 : f8 a2 07 a0 00 18 20 f0   83
2249 : ff a9 0f 8d 76 99 60 1b   7e
2251 : 43 48 1b 4e 09 1b 40 00   b3
2259 : ad 62 99 d0 4b 20 cc ff   a1
2261 : a9 04 20 c3 ff a9 04 aa   3f
2269 : a0 00 20 ba ff a9 00 20   f6
2271 : bd ff 20 c0 ff ad 79 99   d4
2279 : c9 50 f0 0a a2 04 20 c9   46
2281 : ff 90 0b 4c ad 9a a2 04   57
2289 : 20 c9 ff 20 7f 9a a0 00   e1
2291 : b9 24 9a 20 d2 ff c8 c0   d9
2299 : 09 d0 f5 a9 04 20 c3 ff   0d
22a1 : 20 cc ff a9 01 8d 62 99   96
22a9 : 18 60 ad 0d dd ad 00 dd   05
22b1 : 09 04 8d 00 dd a9 ff 8d   66
22b9 : 01 dd ad 00 dd 29 fb 8d   46
22c1 : 00 dd ea 09 04 8d 00 dd   f4
22c9 : a0 ff 88 d0 fd ea ad 0d   ad
22d1 : dd 29 10 f0 01 60 68 68   eb
22d9 : a9 04 20 c3 ff 20 cc ff   39
22e1 : 38 60 12 20 44 20 41 20   5d
22e9 : 54 20 45 20 56 20 2d 20   fe
22f1 : 42 20 45 20 46 20 45 20   53
22f9 : 48 20 4c 20 45 20 92 00   08
2301 : 0e 12 20 41 20 92 20 3d   da
2309 : 20 41 45 4e 44 45 52 4e   39
2311 : 20 20 20 20 20 20 20 20   11
2319 : 20 12 20 43 20 92 20 3d   44
2321 : 20 43 4f 4c 4f 52 20 20   89
2329 : 0d 0d 00 12 20 44 20 92   c9
2331 : 20 3d 20 44 52 55 43 4b   f4
2339 : 45 4e 20 20 20 20 20 20   75
2341 : 20 20 20 12 20 45 20 92   8e
2349 : 20 3d 20 45 49 4e 47 41   5f
2351 : 42 45 0d 0d 00 12 20 46   b8
2359 : 20 92 20 3d 20 46 4f 52   88
2361 : 4d 41 54 49 45 52 45 4e   26
2369 : 20 20 20 20 20 12 20 48   49
2371 : 20 92 20 3d 20 48 41 52   78
2379 : 44 43 4f 50 59 0d 0d 00   6f
2381 : 12 20 4c 20 92 20 3d 20   1a
2389 : 4c 4f 45 53 43 48 45 4e   61
2391 : 20 20 20 20 20 20 20 20   91
2399 : 12 20 51 20 92 20 3d 20   73
23a1 : 51 55 49 54 0d 0d 00 12   d7
23a9 : 20 53 20 92 20 3d 20 53   e0
23b1 : 55 43 48 45 4e 20 20 20   09
23b9 : 20 20 20 20 20 20 20 12   9d
23c1 : 20 57 20 92 20 3d 20 57   02
23c9 : 41 45 48 4c 45 4e 0d 0d   5e
23d1 : 00 12 2b 2f 2d 92 20 3d   ed
23d9 : 20 42 4c 41 45 54 54 45   28
23e1 : 52 4e 20 20 20 20 20 20   2a
23e9 : 20 12 20 5a 20 92 20 3d   f7
23f1 : 20 5a 53 20 57 45 43 48   54
23f9 : 0d 0d 12 20 5f 20 92 20   97
2401 : 3d 20 41 4c 50 48 41 42   f9
2409 : 2e 41 55 53 47 41 42 45   aa
2411 : 0d 0d 00 20 20 20 20 20   6d
2419 : 12 20 47 20 4c 20 4f 20   54
2421 : 42 20 41 20 4c 20 45 20   e3
2429 : 20 42 20 45 20 46 20 45   5a
2431 : 20 48 20 4c 20 45 20 0d   ce
2439 : 0d 00 12 20 c1 20 92 20   77
2441 : 3d 20 41 45 4e 44 45 52   48
2449 : 4e 20 20 20 20 20 20 20   77
2451 : 20 20 12 20 c4 20 92 20   e2
2459 : 3d 20 44 52 55 43 4b 45   29
2461 : 4e 0d 0d 00 12 20 cc 20   0f
2469 : 92 20 3d 20 4c 4f 45 53   5a
2471 : 43 48 45 4e 20 20 20 20   b7
2479 : 20 20 20 20 12 20 cd 20   4f
2481 : 92 20 3d 20 4d 4f 4e 49   92
2489 : 54 4f 52 0d 0d 0d 00 20   35
2491 : 20 20 20 20 20 20 42 49   6c
2499 : 54 54 45 20 45 49 4e 45   cf
24a1 : 20 54 41 53 54 45 20 44   1f
24a9 : 52 55 45 43 4b 45 4e 00   78
24b1 : 20 44 e5 20 fc 9c a2 00   b0
24b9 : a0 06 18 20 f0 ff a9 b7   8c
24c1 : a0 9a 20 1e ab a2 03 a0   97
24c9 : 00 18 20 f0 ff a9 d5 a0   e1
24d1 : 9a 20 1e ab a9 00 a0 9b   cd
24d9 : 20 1e ab a9 2a a0 9b 20   7f
24e1 : 1e ab a9 55 a0 9b 20 1e   8d
24e9 : ab a9 7c a0 9b 20 1e ab   26
24f1 : a9 a6 a0 9b 20 1e ab a9   7e
24f9 : e8 a0 9b 20 1e ab a9 0f   20
2501 : a0 9c 20 1e ab a9 39 a0   e9
2509 : 9c 20 1e ab a9 64 a0 9c   2c
2511 : 20 1e ab 20 e4 ff f0 fb   39
2519 : 60 5a 45 49 43 48 45 4e   49
2521 : 53 41 54 5a 20 33 00 a9   64
2529 : 0d a2 ee a0 9c 20 bd ff   19
2531 : a9 01 20 c3 ff a9 01 a2   72
2539 : 08 a0 02 20 ba ff 20 c0   c4
2541 : ff b0 17 a9 00 a8 a2 08   73
2549 : 20 d5 ff b0 0d a9 13 8d   ef
2551 : 18 d0 a9 01 20 c3 ff 4c   15
2559 : 44 e5 20 69 9d 4c 44 e5   de
2561 : 48 a4 d3 98 d0 06 20 01   c3
2569 : e7 4c 5c 9d 88 84 d3 20   7d
2571 : 24 ea ec 41 80 f0 10 e8   0f
2579 : c8 b1 d1 88 91 d1 c8 b1   cd
2581 : f3 88 91 f3 c8 d0 eb a9   b1
2589 : 20 91 d1 ad 86 02 91 f3   42
2591 : 68 aa ca 60 20 92 9d a9   6d
2599 : 0f a2 08 a0 0f 20 ba ff   ec
25a1 : a9 00 20 bd ff 20 c0 ff   0e
25a9 : 20 cc ff a2 0f 20 c6 ff   91
25b1 : a0 00 20 cf ff 99 00 cb   b8
25b9 : c8 24 90 50 f5 20 cc ff   55
25c1 : a9 0f 20 c3 ff 60 20 ba   6b
25c9 : ff a9 00 20 bd ff 20 c0   7f
25d1 : ff 20 cc ff a2 0f 20 c6   c4
25d9 : ff a0 00 20 cf ff 99 00   90
25e1 : cb c8 24 90 50 f5 20 cc   fa
25e9 : ff a9 0f 20 c3 ff 60 20   83
25f1 : cc ff a9 0f 20 c3 ff 60   ea
25f9 : 60 00 00 00 00 00 00 00   5a
2601 : 00 00 00 00 00 00 00 00   02
2609 : 00 00 00 00 00 00 00 00   0a
2611 : 00 00 00 00 00 00 00 00   12
2619 : 00 00 00 00 00 00 00 00   1a
2621 : 00 00 00 00 00 00 00 00   22
2629 : 00 00 00 4c 07 9e 28 18   e9
2631 : 28 ff 20 31 87 20 2d 9a   ea
2639 : 90 03 4c be 92 20 77 9e   7b
2641 : a9 00 85 fd a9 04 85 fe   da
2649 : a2 04 a0 00 ad 04 9e 8d   a6
2651 : 06 9e ad 03 9e 8d 05 9e   1a
2659 : ad 05 9e d0 10 a9 0d 20   0d
2661 : d2 ff ad 03 9e 8d 05 9e   a6
2669 : ce 06 9e 30 20 b1 fd 85   7b
2671 : d7 29 3f 06 d7 24 d7 10   8c
2679 : 02 09 80 70 02 09 40 20   d8
2681 : d2 ff ce 05 9e c8 d0 d0   bc
2689 : e6 fe ca d0 cb a9 1b 20   72
2691 : d2 ff a9 40 20 d2 ff 20   ae
2699 : cc ff a9 01 20 c3 ff 4c   a8
26a1 : ca 86 a9 01 85 b8 a9 04   06
26a9 : 85 ba a9 07 85 b9 a9 00   a3
26b1 : 85 b7 20 c0 ff a2 01 20   8b
26b9 : c9 ff 60 00 00 a5 cb c9   8a
26c1 : 40 d0 08 a9 cb 8d e5 ce   ff
26c9 : 4c 31 ea a5 cb cd e5 9e   1d
26d1 : d0 03 4c 31 ea a9 0f 8d   af
26d9 : 18 d4 a9 03 8d 05 d4 a9   ce
26e1 : f2 8d 06 d4 a9 2a 8d 01   da
26e9 : d4 a9 18 8d 00 d4 a9 21   d9
26f1 : 8d 04 d4 a0 ff a2 50 ca   b5
26f9 : d0 fd 88 d0 f8 a9 00 8d   fc
2701 : 04 d4 8d 05 d4 8d 06 d4   ef
2709 : a5 cb 8d e5 9e 60 31 ea   3b
2711 : 40 1f ff 1f ff f8 3f ff   89
2719 : fc 60 00 06 ce 45 13 c9   fd
2721 : 45 93 c9 45 53 ce 45 53   b2
2729 : ca 45 33 c9 45 13 c9 39   22
2731 : 13 c0 00 03 dd ce 73 d0   c9
2739 : 91 4b d0 91 4b dc 91 73   9f
2741 : c4 91 43 c4 91 43 dc 8e   fb
2749 : 43 60 00 06 3f ff fc 1f   a3
2751 : ff f8 00 a0 63 b9 e8 9e   c5
2759 : 99 40 03 88 10 f7 a9 1b   82
2761 : 8d 04 d0 a9 85 8d 05 d0   d4
2769 : a9 0d 8d fa 07 a9 04 8d   44
2771 : 15 d0 ad 86 02 8d 29 d0   fd
2779 : 60 00 a9 0f 8d 18 d4 a9   66
2781 : 20 8d 05 d4 a9 f0 8d 06   a8
2789 : d4 a9 04 8d 01 d4 a9 16   6e
2791 : 8d 00 d4 a9 21 8d 04 d4   c1
2799 : a0 ff a2 ff ca d0 fd 88   1e
27a1 : d0 f8 a9 00 8d 04 d4 8d   bf
27a9 : 05 d4 8d 06 d4 60 29 43   b8
```

Listing 1. »Datev« (Schluß)
Das Programm muß mit dem MSE eingegeben werden. Hinweise zum Abtippen finden Sie auf Seite 100.

Turbotape-Copy – Ein Programm für Datasetten-Fans

Turbotape-Copy versetzt Sie in die Lage, alle Basic- und Maschinenprogramme zügig und problemlos auf andere Kassetten zu kopieren.

Mit dem Programm »Turbotape-Copy« (Listing) lassen sich alle Programme, die im Bereich von $0801 (Basic-Start) bis $C34F liegen, von einer zur anderen Kassette kopieren. Die zu kopierenden Programme müssen im Turbotape-Format vorliegen und werden auch in diesem Format gespeichert. Gestartet wird das Programm mit SYS 50944 ($C700).

Es erscheint der Titel mit der Aufforderung, die Kassette mit dem Programm, das kopiert werden soll einzulegen.

Nach dem Drücken der <RETURN>-Taste macht das Programm mit der Meldung
»PRESS PLAY ON TAPE«
darauf aufmerksam, die <PLAY>-Taste zu drücken. Das Programm wird nun an die Originaladresse geladen. Anschließend erscheint die Aufforderung, die Zielkassette einzulegen, auf die das Programm übertragen werden soll. Um zu der Meldung
»PRESS PLAY & RECORD ON TAPE«
zu kommen, ist erneut die <RETURN>-Taste zu drücken. Nach dem Speichervorgang wird die Frage gestellt, ob noch weitere Programme kopiert werden sollen, worauf mit <N> ins Basic zurückgesprungen oder mit <J> ein weiteres Programm kopiert werden kann. (Bausch/Klein/ah)

```
Name : turbotapecopy        c350 c810
-------------------------------------
c350 : a9 5b 8d 08 03 a9 c3 8d  b3
c358 : 09 03 60 20 73 00 f0 04  02
c360 : c9 5f f0 03 4c e7 a7 20  58
c368 : 73 00 c9 53 f0 0b c9 4c  df
c370 : f0 10 c9 56 f0 15 4c 08  9e
c378 : af 20 73 00 20 f0 c3 4c  45
c380 : ae a7 20 73 00 20 e0 c4  86
c388 : 4c ae a7 20 73 00 20 e3  99
c390 : c4 4c ae a7 00 00 00 00  1b
c398 : 00 00 00 00 00 00 00 00  99
c3a0 : 53 55 50 45 52 4d 4f 4e  c4
c3a8 : 20 36 34 0d 00 00 00 00  92
c3b0 : 00 00 00 00 00 00 00 00  b1
c3b8 : 00 00 00 00 00 00 00 00  b9
c3c0 : 00 00 00 00 00 00 00 00  c1
c3c8 : 00 00 00 00 00 00 00 00  c9
c3d0 : 00 00 00 00 00 00 00 00  d1
c3d8 : 00 00 00 00 00 00 00 00  d9
c3e0 : 00 00 00 00 00 00 00 00  e1
c3e8 : 00 00 00 00 00 00 00 00  e9
c3f0 : a2 05 86 ab 20 d4 e1 a2  a1
c3f8 : 04 b5 2a 95 ab ca d0 f9  5c
c400 : 20 38 f8 20 8f f6 20 7d  ab
c408 : c4 20 91 c4 a5 b9 18 69  35
c410 : 01 ca 20 b1 c4 a2 08 b9  aa
c418 : ac 00 20 b1 c4 a2 06 c8  0e
c420 : c0 05 ea d0 f2 a0 00 a2  b1
c428 : 04 b1 bb c4 b7 90 03 a9  ec
c430 : 20 ca 20 b1 c4 a2 05 c8  fb
c438 : c0 bb d0 ed a9 02 85 ab  e0
c440 : 20 91 c4 98 20 b1 c4 84  19
c448 : d7 a2 07 ea b1 ac 20 b1  f4
c450 : c4 a2 03 e6 ac d0 04 e6  32
c458 : ad ca ca a5 ac c5 ae a5  d1
c460 : ad e5 af 90 e7 ea a5 d7  1a
c468 : 20 b1 c4 a2 07 88 d0 f6  cc
c470 : c8 84 c0 58 18 a9 00 8d  9f
c478 : a0 02 4c 93 fc a0 00 84  7d
c480 : c0 ad 11 d0 29 ef 8d 11  e0
c488 : d0 ca d0 fd 88 d0 fa 78  9d
c490 : 60 a0 00 a9 02 20 b1 c4  e7
c498 : a2 07 88 c0 09 d0 f4 a2  28
c4a0 : 05 c6 ab d0 ee 98 20 b1  a5
c4a8 : c4 a2 07 88 d0 f7 ca ca  1e
c4b0 : 60 85 bd 45 d7 85 d7 a9  47
c4b8 : 08 85 a3 06 bd a5 01 29  8c
c4c0 : f7 20 d3 c4 a2 11 ea 09  c5
c4c8 : 08 20 d3 c4 a2 0e c6 a3  6b
c4d0 : d0 e9 60 ca d0 fd 90 05  4f
c4d8 : a2 0b ca d0 fd 85 01 60  9d
c4e0 : a2 00 2c a2 01 a4 2b a5  0f
c4e8 : 2c 86 0a 86 93 84 c3 85  22
c4f0 : c4 20 d4 e1 20 fd c4 20  7b
c4f8 : 7a e1 4c 74 a4 20 61 c5  61
c500 : a5 ab c9 02 f0 08 c9 01  a6
c508 : d0 f3 a5 b9 f0 0a [illegible] [illegible]  01
c510 : 03 85 c3 ad 3d 03 85 c4  08
c518 : 20 50 f7 20 e4 ff f0 fb  6c
c520 : 20 2c a8 a4 b7 f0 0b 88  55
c528 : b1 bb d9 41 03 d0 ce 98  78
c530 : d0 f5 84 90 20 d2 f5 ad  fa
c538 : 3e 03 38 ed 3c 03 08 18  f0
c540 : 65 c3 85 ae ad 3f 03 65  6a
c548 : c4 28 ed 3d 03 85 af 20  9f
c550 : 76 c5 a5 bd 45 d7 05 90  12
c558 : f0 04 a9 ff 85 90 4c a9  16
c560 : f5 20 af c5 c9 00 f0 f9  5e
c568 : 85 ab 20 dd c5 91 b2 c8  cc
c570 : c0 c0 d0 f6 f0 2d 20 af  fc
c578 : c5 20 dd c5 c4 93 d0 02  ad
c580 : 91 c3 d1 c3 f0 02 86 90  3a
c588 : 45 d7 85 d7 e6 c3 d0 02  e9
c590 : e6 c4 a5 c3 c5 ae a5 c4  ac
c598 : e5 af 90 dd 20 dd c5 20  7d
c5a0 : 7d c4 c8 84 c0 58 18 a9  c4
c5a8 : 00 8d a0 02 4c 93 fc 20  6d
c5b0 : 17 f8 20 7d c4 84 d7 a9  1e
c5b8 : 07 8d 06 dd a2 01 20 f0  58
c5c0 : c5 26 bd a5 bd c9 02 d0  90
c5c8 : f5 a0 09 20 dd c5 c9 02  8b
c5d0 : f0 f9 c4 bd d0 e8 20 dd  36
c5d8 : c5 88 d0 f6 60 a9 08 85  73
c5e0 : a3 20 f0 c5 26 bd ea ea  5a
c5e8 : ea c6 a3 d0 f4 a5 bd 60  6c
c5f0 : a9 10 2c 0d dc f0 fb ad  ef
c5f8 : 0d dd 8e 07 dd 48 a9 19  71
c600 : 8d 0f dd 68 4a 4a 60 00  12
c608 : 00 00 00 00 00 00 00 00  09
c610 : c5 26 bd a5 bd c9 02 d0  e0
c618 : f5 a0 09 20 dd c5 c9 02  db
c620 : f0 f9 c4 bd d0 e8 20 dd  86
c628 : c5 88 d0 f6 60 a9 08 85  c3
c630 : a3 20 f0 c5 26 bd ea ea  aa
c638 : ea c6 a3 d0 f4 a5 bd 60  bc
c640 : a9 10 2c 0d dc f0 fb ad  3f
c648 : 0d dd 8e 07 dd 48 a9 19  c1
c650 : 8d 0f dd 68 4a 4a 60 00  62
c658 : 00 00 00 00 00 00 00 00  59
c660 : a4 00 ff 00 00 00 00 00  04
c668 : ff 00 00 00 00 ff 00 ff  67
c670 : 00 10 00 01 df 00 00 00  97
c678 : 16 01 00 01 7b 00 f7 00  c7
c680 : 00 00 00 b6 00 00 00 00  57
c688 : 00 00 00 00 00 00 00 1a  bd
c690 : 00 da 00 00 00 00 00 df  bd
c698 : 5f de a5 00 ff 00 00 ff  d0
c6a0 : 00 e3 00 00 00 ff 00 00  92
c6a8 : 00 df 00 01 18 3e bf 01  2d
c6b0 : 00 00 ff 00 00 00 00 00  b0
c6b8 : 00 00 00 00 00 00 00 04  c1
c6c0 : 00 00 00 00 00 db 00 00  9f
c6c8 : 00 00 00 00 00 4a 00 00  1b
c6d0 : 00 ff ff 59 ff 18 ff ff  bc
c6d8 : ff ff ff ff ff fe 9e 7f  48
c6e0 : ff ff ff ff bf ff ff ff  db
c6e8 : ff ff ff ff ff 08 ff de  e5
c6f0 : ff ff 10 10 9e ff ff ff  df
c6f8 : ff ff ff ff 5b ff ff 7e  a9
c700 : a0 00 b9 88 c7 20 d2 ff  e8
c708 : c8 c0 4d d0 f5 20 73 c7  5b
c710 : a0 00 84 b7 84 0a 84 93  9a
c718 : c8 84 b9 20 fd c4 a0 00  1d
c720 : b9 d5 c7 20 d2 ff c8 c0  8b
c728 : 19 d0 f5 20 73 c7 c6 01  bd
c730 : a4 ab 88 84 b9 a2 03 bd  95
c738 : 3c 03 95 ac ca 10 f8 86  0f
c740 : b7 a2 05 86 ab a9 41 85  73
c748 : bb a9 03 85 bc a2 00 86  37
c750 : 9d 20 00 c4 e6 01 a0 00  8f
c758 : b9 f7 c7 20 d2 ff c8 c0  d4
c760 : 17 d0 f5 20 e4 ff c9 4e  73
c768 : f0 1d c9 4a d0 f5 a0 34  4a
c770 : 4c 02 c7 a0 00 b9 ee c7  dc
c778 : 20 d2 ff c8 c0 09 d0 f5  9e
c780 : 20 e4 ff c9 0d d0 f9 60  4b
c788 : 93 11 11 2a 2a 2a 54 55  1d
c790 : 52 42 4f 54 41 50 45 43  94
c798 : 4f 50 59 20 2a 2a 2a 0d  20
c7a0 : 0d 42 59 20 4d 20 4b 4c  c4
c7a8 : 45 49 4e 20 2b 20 4d 20  53
c7b0 : 42 41 55 53 43 48 20 31  ac
c7b8 : 39 38 36 0d 0d 51 55 45  78
c7c0 : 4c 4c 4b 41 53 53 45 54  bb
c7c8 : 54 45 20 45 49 4e 4c 45  32
c7d0 : 47 45 4e 21 20 0d 0d 5a  c5
c7d8 : 49 45 4c 4b 41 53 53 45  c7
c7e0 : 54 54 45 20 45 49 4e 4c  24
c7e8 : 45 47 45 4e 21 20 12 52  ec
c7f0 : 45 54 55 52 4e 92 0d 0d  c7
c7f8 : 4e 4f 43 48 20 45 49 4e  b6
c800 : 45 20 4b 4f 50 49 45 3f  f5
c808 : 20 4a 2f 4e 20 20 20 20  a7
```

Listing. »Turbotape-Copy« – Ein Programm zum Kopieren von Files im Turbotape-Format. Beachten Sie bitte die Eingabehinweise auf Seite 100.

79 more Blocks free!

Wer seine Disketten bis aufs letzte Byte mit Daten füllen möchte, wird sich über das nächste Programm freuen. Mit ihm lassen sich die bisher ungenutzten Tracks 36 bis 40 zur Datenspeicherung nutzen.

Laut Murphys Gesetzen ist eine Diskette genau in dem Augenblick voll, in dem man ein paar wenige Daten darauf speichern möchte. So muß man dann oft seine Programme wegen einiger weniger Blocks auf zwei Diskettenseiten verteilen.

Doch dem kann man abhelfen. In die Floppy 1541 ist ein 40-Track-Laufwerk eingebaut, das DOS nutzt aber nur 35 Tracks aus. Mit Hilfe einiger Manipulationen im Floppybetriebssystem ist es möglich, auch auf den Spuren 36 bis 40 Daten unterzubringen.

In Listing 1 finden Sie eine Basic-Erweiterung, mit der sich recht einfach Daten auf diesen Spuren ablegen und wieder lesen lassen. Die Routinen sind sehr flexibel gehalten, damit erfahrene Programmierer sie für den Eigenbedarf umändern können. Deswegen finden Sie in Listing 2 auch einen vollständig dokumentierten Source-Code der Basic-Erweiterung.

Es soll nicht verschwiegen werden, daß diese Erweiterung einige Nachteile hat: Sie ist nicht sehr einfach zu bedienen, und der Benutzer muß auf einige Punkte genau achtgeben, da es sonst zu extremen Fehlfunktionen des Floppy-Laufwerks kommen kann! Außerdem gibt es kein Directory und keine BAM für die Spuren 36 bis 40. Sie müssen sich selber merken, wo Sie welche Daten gespeichert haben. Es empfiehlt sich also, nur einige wichtige Daten dort unterzubringen.

5 * $FF	$0E	$XX	$XX	$00	9 * $XX
(Sync)	Header-kenn-zeichen	Track	Sektor	Leer	Lücke

5 * $FF	$0B	255*$XX	$XX	$XX	7 * $XX
(Sync)	Block-kenn-zeichen	255 Daten-byte	interne Prüf-summe	normale Prüf-summe	Lücke

Bild 1. Das neue Format auf den Spuren 36 bis 40

Das Schreiben der Daten ist aufgrund eines geänderten Formats recht langsam. Eben wegen diesem geänderten Format sind die Spuren 36 bis 40 auch nicht mit normalen Kopierprogrammen kopierbar – da muß man schon mit schweren Geschützen wie Turbonibbler und ähnlichen auffahren. Wie das Diskettenformat genau aussieht, zeigt Bild 1.

Fünf neue Befehle

Die Basic-Erweiterung wird mit »LOAD"name",8,1« geladen und mit »SYS 50000« gestartet. Danach stehen Ihnen folgende fünf neuen Befehle zur Verfügung (Die eckigen Klammern gehören zum Befehl und müssen mit eingegeben werden.):

[NEW] – formatiert auf einer Diskette die Spuren 36-40. Dies muß gemacht werden, bevor irgendwelche Daten geschrieben werden. Daten, die sich auf den Spuren 1 bis 35 befinden, werden nicht geändert. Sie können also Disketten auch nachträglich mit den neuen Spuren ausstatten.

Ab sofort stehen Ihnen 79 Blöcke, numeriert von 0 bis 78, zur Verfügung. Auf den Spuren 36 bis 39 befinden sich jeweils 16 Blöcke, auf Spur 40 nur 15. Diese Blöcke können nur über ihre Blocknummer und nicht über Track/Sektor-Angaben angesprochen werden!

[SAVE anfang, ende, blockstart] – Mit diesem Befehl können Sie einen Speicherbereich auf die neuen Spuren schreiben. Anfang und Ende bezeichnen dabei die Start- und Endadresse des zu schreibenden Breichs. Mit Blockstart geben Sie den Block an, bei dem die Aufzeichnung beginnen soll. Nach dem Speichervorgang können Sie mit PEEK(139) erfragen, bis zu welchem Block geschrieben wurde.

[LOAD anfang, blockstart] – Das Gegenstück zu SAVE. Mit diesem Befehl wird ein Speicherbereich ab dem Block der Nummer Blockstart an die Adresse anfang geladen.

[GET stringvariable, block] - Mit diesem Befehl lassen sich die 255 Byte eines Blocks in eine beliebige Stringvariable lesen.

[PRINT stringvariable, block] – Dieser Befehl schreibt die Daten aus einem String in den angegebenen Block. Die Länge des Strings muß genau 255 Zeichen betragen, ansonsten kann »Müll« im Block auf der Diskette stehen!

Folgende Fehlermeldungen werden von den neuen Befehlen ausgegeben:

Division by Zero – Sie haben eine kleinere End- als Anfangsadresse angegeben.

Illegal Quantity – Es sollte auf einen Block mit einer Nummer größer 78 zurückgegriffen werden.

Overflow – Sie wollten mehr als 79 Blöcke in einem Stück speichern.

String too long – Der String, den Sie in einen Block schreiben wollten, hat nicht exakt 255 Zeichen (meistens ist der String dann also zu kurz, aber dieses Paradoxon soll nicht weiter stören).

Wenn nötig, läßt sich die Basic-Erweiterung mit »SYS 50011« wieder ausschalten. Mit »POKE 49799,x« kann man die Adresse der anzusprechenden Floppy auf andere Werte ändern, der Standardwert ist 8.

Wenn auf eine nicht mit [NEW] formatierte Disk zugegriffen werden soll, stürzt das Programm unweigerlich ab! Dies geschieht ebenso, wenn keine Diskette im Laufwerk liegt oder gar kein Diskettenlaufwerk angeschlossen ist!

Anwendungsbeispiele

Damit Sie sich im Umgang mit den neuen Basic-Befehlen üben können, folgen nun einige Anwendungsbeispiele. Als erstes sollten Sie eine Diskette mit [NEW] formatieren.

Auf der Diskette befinden sich noch keine Daten, also können wir ab Block 0 mit dem Schreiben von Daten beginnen. Mit [SAVE 1024, 2023, 0] speichern wir als erstes einmal den Bildschirmspeicher. Mit »?PEEK(139)« erfahren wir, daß der letzte beschriebene Block die Nummer 3 ist. Um also einen weiteren Bildschirm zu speichern, müssen wir [SAVE 1024, 2023, 4] eingeben, da ab dem 4. Block freie Blöcke vorhanden sind. Um den ersten Bildschirm wieder zu laden, geben Sie [LOAD 1024, 0] ein.

Komplizierter wird es da schon, wenn Sie ein Basic-Programm speichern wollen. Hier sind leider eine Reihe von POKEs notwendig, um alle Vektoren wieder geradezubiegen.

```
1 REM BASIC-DEMO ZUR [GET+PRINT]
2 REM                              :
3 REM          GERMANO CARONNI
4 REM          GREUBSTELSTR.10
5 REM         CH-5430 WETTINGEN
6 REM                              :
8 REM DEMO ZU TRACK40,(MUSS IM SPEICHER SEIN)
9 POKE53280,0:POKE53281,0:PRINT""
10 SYS50000
20 INPUT"BLOCK (0-78)";B:IFB<0ORB>78THEN25
30 INPUT"TEXT";A$:B$=CHR$(0):FORI=1TO7
31 B$=B$+B$:NEXT:B$=B$+LEFT$(B$,127)
40 A$=A$+LEFT$(B$,255-LEN(A$))
50 [PRINTA$,B]
60 PRINT"TEXT IST GESPEICHERT  --  TASTE"
70 WAIT198,1:POKE198,0
80 [GETB$,B]
90 PRINT"DER NEUE TEXT :":PRINTB$
100 PRINT"DER ALTE TEXT :":PRINTA$
110 PRINT"";A$=B$

READY.
```

Listing 3. Ein kleines Beispielprogramm zur Anwendung der neuen Befehle

Das Speichern wird mit folgender Befehlsfolge ausgelöst: [SAVE 2049, PEEK(45)+256*PEEK(46), block]

Um das Programm später wieder zu laden, genügt ein [LOAD 2049, block]. Allerdings müssen jetzt einige Vektoren geradegebogen werden. Dazu geben Sie direkt nach dem Laden folgende zwei Zeilen ein:

POKE 46,159 : CLR : L=(PEEK(139)-block) *254 +PEEK(141)

POKE 45, (L+2049) AND 255 : POKE 46, (L+2049)/256 : CLR

Dieser Aufwand ist zwar nicht unerheblich, erleichterte aber die Programmierung der Basic-Erweiterung ungemein. Normalerweise wird man ja auch keine Basic-Programme auf den Spuren 36 bis 40 ablegen.

Unser Listing 3 zeigt Ihnen ein kleines Beispiel zur Anwendung der [PRINT]- und [GET]-Befehle.

Laden ohne Erweiterung

Manchmal ist es notwendig, daß die Daten auf den neuen Spuren von einem Programm gelesen werden müssen, die Basic-Erweiterung aber nicht aktiv sein darf. Für diesen Fall gibt es im Listing 4 die Laderoutine auch lose und relokatibel. Die Anwendung ist etwas kompliziert, dadurch kann die Laderoutine aber sehr einfach in eigene Programme eingebaut werden. Dies sollten Sie aber nur versuchen, wenn Sie schon Erfahrung mit Maschinensprache haben.

Zur Erklärung: Nachdem Sie die Laderoutine mit »LOAD "name",8,1« geladen haben, können Sie sie mit einem Monitor in beliebige Speicherbereiche legen. Um die Adressen anzupassen, müssen Sie die Startadresse der Laderoutine im LO/HI-Format in 254, 254 schreiben. Der Anfangsblock muß in Speicherzelle 139 geschrieben werden, die Ladeadresse im LO/HI-Format in die Zellen 251 und 252.

Danach wird die Laderoutine mit »SYS startadresse+510« aufgerufen. Soll ein zweites Mal geladen werden, genügt die Angabe von neuer Ladeadresse und Anfangsblock und der Aufruf mit »SYS startadresse«.

Nach dem Ladevorgang kann in der Speicherzelle 2 geprüft werden, ob alles geklappt hat. Dann steht dort eine 0. Trat während des Ladens ein Fehler auf, ist dort eine 14 anzufinden.

(Germano Carroni/bs)


```
Name : track40               c000 c750
--------------------------------------
c000 : ad 00 1c 29 9f 8d 00 1c  78
c008 : a0 04 84 31 a0 00 84 30  6e
c010 : 84 34 a9 0e 85 52 ad fe  7a
c018 : 07 85 53 84 54 a9 00 85  e5
c020 : 55 98 48 20 d0 f6 68 a8  8f
c028 : c8 cc fd 07 d0 e4 a5 34  ea
c030 : 8d ff 07 a9 0d 85 47 a0  0f
c038 : 00 98 99 00 03 c8 d0 fa  9a
c040 : 85 30 85 34 a9 03 85 31  f0
c048 : 20 e9 f5 85 3a 20 8f f7  5e
c050 : 20 0e fe a2 00 a0 05 a9  f8
c058 : ff 50 fe b8 8d 01 1c 88  b8
c060 : d0 f7 a0 05 bd 00 04 e8  b2
c068 : 50 fe b8 8d 01 1c 88 d0  cc
c070 : f3 a0 09 a9 55 50 fe b8  70
c078 : 8d 01 1c 88 d0 f7 a0 05  f7
c080 : a9 ff 50 fe b8 8d 01 1c  51
c088 : 88 d0 f7 a0 bb b9 00 01  16
c090 : 50 fe b8 8d 01 1c c8 d0  f5
c098 : f4 b9 00 03 50 fe b8 8d  c4
c0a0 : 01 1c c8 d0 f4 a0 07 a9  bf
c0a8 : 55 50 fe b8 8d 01 1c 88  5e
c0b0 : d0 f7 ec ff 07 d0 9e 20  69
c0b8 : 00 fe a9 07 85 47 4c 9e  84
c0c0 : fd a9 10 8d fd 07 a2 24  32
c0c8 : 8e fe 07 86 0a a9 e0 85  e4
c0d0 : 02 a5 02 30 fc e8 e0 28  16
c0d8 : 90 ee ce fd 07 e0 29 d0  10
c0e0 : e7 60 ad 00 1c 29 9f 8d  07
c0e8 : 00 1c a9 00 a8 18 79 00  92
c0f0 : 04 c8 c0 ff d0 f7 8d ff  8b
c0f8 : 04 a9 0d 85 47 a9 04 85  a2
c100 : 31 a9 00 85 30 85 34 20  f7
c108 : e9 f5 85 3a 20 8f f7 a9  46
c110 : 0e 85 52 ad fe 07 85 53  10
c118 : ad ff 07 85 54 a9 00 85  d5
c120 : 55 85 30 85 34 a9 03 85  9c
c128 : 31 20 d0 f6 20 f0 05 50  ba
c130 : fe b8 ad 01 1c d9 00 03  ac
c138 : d0 f2 c8 c0 05 d0 f0 a9  b9
c140 : 07 85 47 a2 06 50 fe b8  80
c148 : ca d0 fa a9 ff 8d 03 1c  1f
c150 : ad 0c 1c 29 1f 09 c0 8d  88
c158 : 0c 1c a9 ff a2 05 8d 01  67
c160 : 1c b8 50 fe b8 ca d0 fa  e7
c168 : a0 bb b9 00 01 50 fe b8  54
c170 : 8d 01 1c c8 d0 f4 b9 00  3a
c178 : 04 50 fe b8 8d 01 1c c8  5e
c180 : d0 f4 50 fe 20 00 fe 20  fc
c188 : f0 05 50 fe b8 ad 01 1c  24
c190 : d9 00 03 d0 f2 c8 c0 05  c7
c198 : d0 f0 d0 90 20 f0 05 a0  05
c1a0 : bb b9 00 01 50 fe b8 4d  d3
c1a8 : 01 1c d0 ee c8 d0 f2 b9  1b
c1b0 : 00 04 50 fe b8 4d 01 1c  d8
c1b8 : d0 e0 c8 d0 f2 4c 9e fd  4c
c1c0 : ad fe 07 85 0a a9 e0 85  db
c1c8 : 02 a5 02 30 fc ea ea ea  cc
c1d0 : ea 60 2c 00 1c 30 fb ad  84
c1d8 : 01 1c b8 a0 00 60 ad 00  e3
c1e0 : 1c 29 9f 8d 00 1c a9 0e  ce
c1e8 : 85 52 ad fe 07 85 53 ad  27
c1f0 : ff 07 85 54 a9 00 85 55  ba
c1f8 : 85 34 85 30 a9 03 85 31  2a
c200 : 20 d0 f6 2c 00 1c 30 fb  65
c208 : ad 01 1c b8 a0 00 50 fe  9d
c210 : b8 ad 01 1c d9 00 03 d0  ae
c218 : ea c8 c0 05 d0 f0 a0 00  4e
c220 : 2c 00 1c 30 fb ad 01 1c  c3
c228 : b8 50 fe b8 ad 01 1c 99  65
c230 : 00 03 c8 d0 f4 a0 ba 50  de
c238 : fe b8 ad 01 1c 99 00 01  ae
c240 : c8 d0 f4 20 e0 f8 a5 38  8e
c248 : c9 0d d0 9a a9 00 85 30  30
c250 : a9 03 85 31 20 e9 f5 c5  b7
c258 : 3a d0 8b a9 00 a8 18 79  ab
c260 : 00 03 c8 c0 ff d0 f7 cd  2e
c268 : ff 03 d0 ed 4c 9e fd ad  e8
c270 : fe 07 85 0a a9 e0 85 02  50
c278 : a5 02 30 fc 60 49 23 32  0b
c280 : 42 2d 50 20 32 20 30 08  66
c288 : a9 e2 2c a9 fc 2c a9 9f  fa
c290 : 48 20 e7 ff a9 01 ae 87  4e
c298 : c2 a0 0f 20 ba ff a9 01  c6
c2a0 : a2 7d a0 c2 20 bd ff 20  b1
c2a8 : c0 ff 90 03 4c f9 e0 a9  58
c2b0 : 02 ae 87 c2 a8 20 ba ff  ba
c2b8 : a9 02 a2 7e a0 c2 20 bd  f7
c2c0 : ff 20 c0 ff b0 e6 20 cc  5c
c2c8 : ff a2 02 20 c6 ff 20 cf  29
c2d0 : ff 20 cc ff a2 01 20 c9  59
c2d8 : ff b0 d1 a0 00 b9 80 c2  0d
c2e0 : 20 d2 ff c8 c0 07 d0 f5  f6
c2e8 : 20 cc ff a2 02 20 c9 ff  0b
c2f0 : b0 ba 68 85 8f a0 00 a5  11
c2f8 : 8f c9 e2 d0 06 b9 00 c0  ee
c300 : 4c 14 c3 c9 fc d0 06 b9  62
c308 : e2 c0 4c 14 c3 c9 9f d0  8b
c310 : 9b b9 de c1 20 d2 ff b0  72
c318 : 93 c8 c4 8f d0 d9 20 cc  28
c320 : ff a9 02 4c c3 ff 4d 2d  ca
c328 : 45 c1 05 4d 2d 57 fe 07  d0
c330 : 02 24 03 23 30 23 31 8d  66
c338 : 29 c3 20 cc ff a2 01 20  3e
c340 : c9 ff a0 00 b9 26 c3 20  4d
c348 : d2 ff c8 c0 05 d0 f5 60  d3
c350 : a9 66 8d 08 03 a9 c3 8d  38
c358 : 09 03 60 a9 e4 a0 a7 8d  3d
c360 : 08 03 8c 09 03 60 20 73  c9
c368 : 00 c9 5b f0 03 4c d4 c5  b3
c370 : 20 73 00 c9 a2 d0 2d 20  29
c378 : 73 00 c9 5d f0 03 4c 08  72
c380 : af 20 88 c2 a9 c1 20 37  51
c388 : c3 20 cc ff a2 01 20 c9  d5
c390 : ff a9 49 20 d2 ff 20 cc  01
c398 : ff a9 01 20 c3 ff 20 73  54
c3a0 : 00 4c ae a7 c9 94 f0 03  72
c3a8 : 4c 17 c5 20 73 00 20 8a  c2
c3b0 : ad 20 f7 b7 84 fb 85 fc  9a
c3b8 : 20 fd ae 20 8a ad 20 f7  0d
c3c0 : b7 84 fd 85 fe 20 fd ae  30
c3c8 : a5 fd 38 e5 fb 85 8b a5  9c
c3d0 : fe e5 fc 85 8c b0 05 a2  58
c3d8 : 14 4c 3a a4 a2 00 a5 8b  0d
c3e0 : 38 e9 fe 85 8b a5 8c e9  69
c3e8 : 00 85 8c 90 03 e8 d0 ee  79
c3f0 : e0 4f 90 05 a2 0f 4c 3a  85
c3f8 : a4 8a 48 20 9e b7 8a 48  5a
c400 : 20 79 00 c9 5d f0 03 4c  18
c408 : 08 af 20 73 00 68 85 8b  cf
c410 : 68 18 65 8b c9 4f 90 05  b2
c418 : a2 0e 4c 3a a4 20 8b c2  1b
c420 : a9 02 ae 87 c2 a8 20 ba  ce
c428 : ff a9 02 a2 35 a0 c3 20  78
c430 : bd ff 20 c0 ff 90 4a 4c  53
c438 : f9 e0 20 cc ff a2 01 20  9c
c440 : c9 ff a0 00 b9 80 c2 20  1c
c448 : d2 ff b0 eb c8 c0 07 d0  14
c450 : f3 4c cc ff a5 8b 4a 4a  11
c458 : 4a 4a 18 69 24 8d 31 c3  f6
c460 : a5 8b 29 0f 8d 32 c3 20  b1
c468 : cc ff a2 01 20 c9 ff a0  8e
c470 : 00 b9 2b c3 20 d2 ff b0  8a
```

```
c478 : be c8 c0 08 d0 f3 4c cc  43
c480 : ff 20 cc ff a2 02 20 c6  0b
c488 : ff 20 cf ff 20 b7 ff f0  2d
c490 : 03 4c 62 a4 20 54 c4 20  de
c498 : 3a c4 a2 02 20 c9 ff a5  b9
c4a0 : fd 38 e5 fb 85 8c a5 fe  03
c4a8 : e5 fc d0 06 a4 8c c0 fe  b0
c4b0 : 90 2d a9 ff 20 d2 ff a0  1b
c4b8 : 00 b1 fb 20 d2 ff b0 b7  f3
c4c0 : c8 c0 fe d0 f4 20 cc ff  45
c4c8 : a9 de 20 37 c3 ea ea e6  dc
c4d0 : 8b a5 fb 18 69 fe 85 fb  cc
c4d8 : 90 02 e6 fc 4c 94 c4 c8  d1
c4e0 : 98 20 d2 ff 84 8d a0 00  74
c4e8 : b1 fb 20 d2 ff c8 c4 8d  6e
c4f0 : d0 f6 20 cc ff a9 de 20  e6
c4f8 : 37 c3 20 cc ff a9 02 20  48
c500 : c3 ff a2 01 20 c9 ff a9  2f
c508 : 49 20 d2 ff 20 cc ff a9  d2
c510 : 01 20 c3 ff 4c ae a7 c9  7e
c518 : 93 f0 03 4c da c5 20 73  b1
c520 : 00 20 8a ad 20 f7 b7 84  32
c528 : fb 85 fc 20 fd ae 20 9e  3c
c530 : b7 e0 4f 90 05 a2 0e 4c  73
c538 : 3a a4 86 8b 20 79 00 20  e5
c540 : 0f c6 ea 20 73 00 20 8e  46
c548 : c2 a9 02 ae 87 c2 a8 20  a7
c550 : ba ff a9 02 a2 33 a0 c3  82
c558 : 20 bd ff 20 c0 ff 90 03  af
c560 : 4c f9 e0 20 cc ff a2 02  40
c568 : 20 c6 ff 20 cf ff 20 b7  dc
c570 : ff f0 03 4c 62 a4 20 54  a6
c578 : c4 a9 91 20 37 c3 20 3a  00
c580 : c4 a2 02 20 c6 ff 20 cf  a6
c588 : ff 85 8d a0 00 20 cf ff  01
c590 : b0 ce 91 fb c8 c4 8d d0  16
c598 : f4 20 cc ff a5 8d c9 ff  bd
c5a0 : d0 18 a5 fb 18 69 fe 85  39
c5a8 : fb 90 02 e6 fc e6 8b a5  c9
c5b0 : 8b c9 4f 90 c1 a2 0e 4c  08
c5b8 : 3a a4 a9 02 20 c3 ff a2  54
c5c0 : 01 20 c9 ff a9 49 20 d2  4f
c5c8 : ff 20 cc ff a9 01 20 c3  b5
c5d0 : ff 4c ae a7 20 79 00 4c  fc
c5d8 : e7 a7 c9 a1 f0 03 4c 97  c1
c5e0 : c6 20 73 00 20 8b b0 85  bf
c5e8 : 49 84 4a a9 ff 20 75 b4  7b
c5f0 : a0 02 b9 61 00 91 49 48  6e
c5f8 : 88 10 f7 20 fd ae 20 9e  9d
c600 : b7 e0 4f 90 05 a2 0e 4c  43
c608 : 3a a4 86 8b 4c 17 c6 c9  d4
c610 : 5d f0 03 4c 08 af 60 20  ef
c618 : 79 00 20 0f c6 20 73 00  b7
c620 : 20 8e c2 68 85 8d 68 85  b6
c628 : fb 68 85 fc a9 02 ae 87  cd
c630 : c2 a8 20 ba ff a9 02 a2  40
c638 : 33 a0 c3 20 bd ff 20 c0  8e
c640 : ff 90 03 4c f9 e0 20 cc  92
c648 : ff a2 02 20 c6 ff 20 cf  a9
c650 : ff 20 b7 ff f0 03 4c 62  6a
c658 : a4 20 54 c4 a9 91 20 37  d0
c660 : c3 20 3a c4 a2 02 20 c6  a3
c668 : ff ea ea ea a0 00 20 cf  1e
c670 : ff b0 d0 91 fb c8 c4 8d  62
c678 : d0 f4 20 cc ff a9 02 20  fa
c680 : c3 ff a2 01 20 c9 ff a9  af
c688 : 49 20 d2 ff 20 cc ff a9  52
c690 : 01 20 c3 ff 4c ae a7 c9  fe
c698 : 99 f0 03 4c 08 af 20 73  59
c6a0 : 00 20 8b b0 48 98 48 20  54
c6a8 : fd ae 20 9e b7 e0 4f 90  b9
c6b0 : 05 a2 0e 4c 3a a4 86 8b  0d
c6b8 : 20 79 00 20 0f c6 20 73  28
c6c0 : 00 68 85 62 68 20 40 c7  ba
c6c8 : c8 b1 61 85 fb c8 b1 61  01
c6d0 : 85 fc 20 8b c2 a9 02 ae  2c
c6d8 : 87 c2 a8 20 ba ff a9 02  45
c6e0 : a2 35 a0 c3 20 bd ff 20  ee
c6e8 : c0 ff 90 03 4c f9 e0 20  85
c6f0 : cc ff a2 02 20 c6 ff 20  1d
c6f8 : cf ff 20 b7 ff f0 03 4c  f2
c700 : 62 a4 20 54 c4 20 3a c4  07
c708 : a2 02 20 c9 ff a0 00 b1  55
c710 : fb 20 d2 ff b0 d6 c8 c0  36
c718 : ff d0 f4 20 cc ff a9 de  f2
c720 : 20 37 c3 20 cc ff a9 02  48
c728 : 20 c3 ff a2 01 20 c9 ff  b6
c730 : a9 49 20 d2 ff 20 cc ff  14
c738 : a9 01 20 c3 ff 4c ae a7  4f
c740 : 85 61 a0 00 b1 61 c9 ff  eb
c748 : f0 05 a2 17 4c 3a a4 60  30
```

Listing 1.
Die Basic-Erweiterung »TRACK40«.
Bitte mit dem MSE (Seite 100) eingeben.

```
100 ;    BASIC-ERWEITERUNG ZUR NUETZUNG DER SPUREN 36-40 AUF DER DISKETTE
110 ;                                                       :
120 ;             GERMANO CARONNI
130 ;             GREUBSTELSTR.10
140 ;            CH-5430 WETTINGEN
150 ;
160 ;                                                       :
900 ;
49152 ".BA  49152 ;ZUERST DREI FLOPPY-ROUTINEN DANN HAUPTPROGRAMM
49152 " LDA  7168 ;START 50000 EINSPRUNGPUNKT VOM INTERPRET 50022
49155 " AND  #159 ;FORMATIEREN (36-40)  --- DC-TIMER FUER SPUR 36 UND HOEHER
49157 " STA  7168 ;EINRICHTEN
49160 " LDY    #4
49162 " STY    49
49164 " LDY    #0
49166 " STY    48 ;ZEIGER  AUF 1024
49168 " STY    52
49170 " LDA   #14
49172 " STA    82 ;HEADER-KENNZEICHEN
49174 " LDA  2046
49177 " STA    83 ;SPUR
49179 " STY    84 ;SEKTOR
49181 " LDA    #0
49183 " STA    85 ;LEER
49185 " TYA
49186 " PHA       ;SEKTOR RETTEN
49187 " JSR 63184 ;DIESE FUENF BYTES IN GCR-CODE WANDELN UND AB $400 ABLEGEN
49190 " PLA       ;(DAS GIBT DIE EINZELNEN HEADER FUER DIE SEKTOREN EINER SPUR)
49191 " TAY       ;SEKTOR WIEDER HOLEN
49192 " INY
49193 " CPY  2045 ;SCHAUEN OB LETZTER SEKTOR
49196 " BNE 49170 ;WENN NICHT DANN DEN NAECHSTEN UMWANDELN
49198 " LDA    52 ;DIE ADRESSE DES LETZTEN BYTES DES LETZTEN SEKTORS +1 HOLEN
49200 " STA  2047 ;UND RETTEN
49203 " LDA   #13
49205 " STA    71 ;DATENBLOCKKENNZEICHEN
49207 " LDY    #0
49209 " TYA
49210 " STA  768,Y
49213 " INY
49214 " BNE 49210 ;DATENBLOCK AB 768 MIT LAUTER NULLEN FUELLEN
49216 " STA    48
49218 " STA    52
49220 " LDA    #3
49222 " STA    49 ;DIE ZEIGER AUF $300 RICHTEN (768)
49224 " JSR 62953 ;PRUEFSUMME BERECHNEN
49227 " STA    58 ;UND SPEICHERN
```


Listing 2.
Der Source-Code zu »TRACK40«.
Bitte nicht eingeben, dient nur zur Dokumentation.

```
49229 " JSR 63375 ;DATENBLOCK IN GCR-KODE WANDELN
49232 " JSR 65038 ;DC AUF SCHREIBEN UND DIE SPUR LOESCHEN
49235 " LDX    #0
49237 " LDY    #5
49239 " LDA  #255
49241 " BVC 49241
49243 " CLV
49244 " STA  7169 ;5*255 ALS HEADER-SYNC(HRONISTATIONSMARKIERUNG) AUF DISKETTE
49247 " DEY
49248 " BNE 49241
49250 " LDY    #5
49252 " LDA  1024,X
49255 " INX
49256 " BVC 49256
49258 " CLV
49259 " STA  7169 ;DANN DIE 5 HEADER-BYTES
49262 " DEY
49263 " BNE 49252
49265 " LDY    #9
49267 " LDA   #85
49269 " BVC 49269
49271 " CLV
49272 " STA  7169 ;9 LEERBYTES ALS LUECKE ABSPEICHERN
49275 " DEY
49276 " BNE 49269
49278 " LDY    #5
49280 " LDA  #255
49282 " BVC 49282
49284 " CLV
49285 " STA  7169   ;DATENBLOCK-SYNC
49288 " DEY
49289 " BNE 49282
49291 " LDY  #187
49293 " LDA   256,Y ;DEN ERSTEN TEIL DES UMGEWANDELTEN BLOCKES AUF DISKETTE
49296 " BVC 49296
49298 " CLV
49299 " STA  7169
49302 " INY
49303 " BNE 49293
49305 " LDA   768,Y ;UND DANN DEN ZWEITEN TEIL
49308 " BVC 49308
49310 " CLV
49311 " STA  7169
49314 " INY
49315 " BNE 49305
49317 " LDY    #7
49319 " LDA   #85
49321 " BVC 49321
49323 " CLV
49324 " STA  7169   ;WIEDER 7 BYTES LUECKE
49327 " DEY
49328 " BNE 49321
49330 " CPX  2047   ;SCHAUEN OB ALLE SEKTOREN EINGERICHTET
49333 " BNE 49237   ;WENN NICHT DANN DEN NAECHSTEN
49335 " JSR 65024   ;DC AUF LESEN SCHALTEN
49338 " LDA    #7
49340 " STA    71   ;DATENBLOCK-KENNZEICHEN WIEDER RICHTEN
49342 " JMP 64926   ;ZURUECK ZUR NORMALEN FLOPPY IRQ-ROUTINE
49345 " LDA   #16   ;EISPRUNGSPUNKT MIT M-E
49347 " STA  2045   ;PRO SPUR 16 SEKTOREN EINRICHTEN
49350 " LDX   #36
49352 " STX  2046   ;SPUR 36
49355 " STX    10   ;DEM IRQ-PRG BEKANNT GEBEN
49357 " LDA  #224
49359 " STA     2   ;IRQ-PRG STARTEN
49361 " LDA     2
49363 " BMI 49361   ;WARTEN AUF FERTIG
49365 " INX
49366 " CPX   #40   ;SCHAUEN OB SCHON BEI SPUR 40
49368 " BCC 49352   ;WENN NICHT NAECHSTE SPUR
49370 " DEC  2045   ;PRO SPUR 15 SEKTOREN
49373 " CPX   #41
49375 " BNE 49352   ;UND DIE SPUR 40 NOCH FORMATIEREN
49377 " RTS         ;ENDE
49378 " LDA  7168   ;EINEN BLOCK AB $400 SPEICHERN (SPUR+SEKTOR IN 2046,2047)
49381 " AND  #159   ;DC-TIMER RICHTEN
49383 " STA  7168
```


Listing 2. Der Source-Code zu »Track40«. Bitte nicht eingeben, dient nur zur Dokumentation.

```
49386 " LDA    #0
49388 " TAY
49389 " CLC
49390 " ADC   1024,Y ;ERSTE (INTERNE) PRUEFSUMME BERECHNEN
49393 " INY
49394 " CPY   #255
49396 " BNE 49389
49398 " STA  1279    ;UND ALS LETZTES DATENBLOCK-BYTE SPEICHERN
49401 " LDA   #13
49403 " STA    71    ;DATENBLOCK KENNZEICHEN RICHTEN
49405 " LDA    #4
49407 " STA    49
49409 " LDA    #0
49411 " STA    48    ;ZEIGER AUF $400
49413 " STA    52
49415 " JSR 62953    ;PRUEFSUMME BERECHNEN
49418 " STA    58    ;SPEICHERN
49420 " JSR 63375    ;DATENBLOCK UMWANDELN IN GCR-KODE
49423 " LDA   #14    ;HEADER AUFBAUEN
49425 " STA    82    ;HEADER-KENNZEICHEN
49427 " LDA  2046
49430 " STA    83    ;SPUR
49432 " LDA  2047
49435 " STA    84    ;SEKTOR
49437 " LDA    #0
49439 " STA    85    ;LEER
49441 " STA    48
49443 " STA    52
49445 " LDA    #3    ;ZEIGER AUF $300
49447 " STA    49
49449 " JSR 63184    ;HEADER IN GCR UMWANDELN (AB 768 IM SPEICHER)
49452 " JSR  1520    ;WARTEN AUF SYNC
49455 " BVC 49455
49457 " CLV
49458 " LDA  7169
49461 " CMP   768,Y ;RICHTIGEN HEADER SUCHEN
49464 " BNE 49452    ;WENN NICHT DANN NAECHSTEN
49466 " INY
49467 " CPY    #5
49469 " BNE 49455
49471 " LDA    #7
49473 " STA    71    ;DATENBLOCK-KENNZEICHEN WIEDER RICHTEN (VERZOEGERUNG)
49475 " LDX    #6
49477 " BVC 49477    ;6 BYTES UEBERLESEN
49479 " CLV
49480 " DEX
49481 " BNE 49477
49483 " LDA  #255
49485 " STA  7171
49488 " LDA  7180
49491 " AND   #31
49493 " ORA  #192
49495 " STA  7180    ;DC AUF SCHREIBEN UMSCHALTEN
49498 " LDA  #255
49500 " LDX    #5
49502 " STA  7169    ;5 BYTES DATEN-SYNC SCHREIBEN
49505 " CLV
49506 " BVC 49506
49508 " CLV
49509 " DEX
49510 " BNE 49506
49512 " LDY  #187
49514 " LDA   256,Y ;ERSTEN TEIL DES BLOCKES
49517 " BVC 49517
49519 " CLV
49520 " STA  7169
49523 " INY
49524 " BNE 49514
49526 " LDA  1024,Y ;UND DANN DEN ZWEITEN TEIL SCHREIBEN
49529 " BVC 49529
49531 " CLV
49532 " STA  7169
49535 " INY
49536 " BNE 49526
49538 " BVC 49538    ;WARTEN DAS DAS LETZTE BYTE FERTIG GESCHRIEBEN IST
49540 " JSR 65024    ;DC AUF LESEN
49543 " JSR  1520    ;WARTEN AUF SYNC
```

```
49546 " BVC 49546
49548 " CLV
49549 " LDA  7169
49552 " CMP   768,Y ;DEN RICHTIGEN HEADER SUCHEN
49555 " BNE 49543
49557 " INY
49558 " CPY    #5
49560 " BNE 49546
49562 " BNE 49452
49564 " JSR  1520
49567 " LDY  #187
49569 " LDA   256,Y ;UND DEN SOEBEN GESCHRIEBENEN BLOCK KONTROLLIEREN
49572 " BVC 49572
49574 " CLV
49575 " EOR  7169
49578 " BNE 49562   ;WENN FEHLER DANN ALLES VON VORNE
49580 " INY
49581 " BNE 49569
49583 " LDA  1024,Y ;ZWEITEN TEIL KONTROLLIEREN
49586 " BVC 49586
49588 " CLV
49589 " EOR  7169
49592 " BNE 49562
49594 " INY
49595 " BNE 49583
49597 " JMP 64926   ;ZURUECK ZUR FLOPPY IRQ-ROUTINE
49600 " LDA  2046
49603 " STA    10   ;SPUR FESTLEGEN
49605 " LDA  #224   ;IRQ-PRG. STARTEN
49607 " STA     2
49609 " LDA     2
49611 " BMI 49609   ;WARTEN AUF FERTIG
49613 " .BY 234 234 ;LUECKE
49615 " .BY 234 234
49617 " RTS         ;ENDE
49618 " BIT  7168   ;WARTEN AUF SYNC
49621 " BMI 49618   ;WENN NOCH KEINE DANN NOCHMAL
49623 " LDA  7169
49626 " CLV         ;DC-PORT FREIMACHEN
49627 " LDY    #0
49629 " RTS
49630 " LDA  7168   ;EINEN BLOCK LESEN UND IN $300 SPEICHERN TR+SE 2046,2047
49633 " AND  #159   ;DC-TIMER RICHTEN
49635 " STA  7168
49638 " LDA   #14
49640 " STA    82   ;HEADER EINRICHTEN
49642 " LDA  2046
49645 " STA    83   ;SPUR
49647 " LDA  2047
49650 " STA    84   ;SEKTOR
49652 " LDA    #0
49654 " STA    85   ;LEER
49656 " STA    52
49658 " STA    48
49660 " LDA    #3   ;ZEIGER AUF $300
49662 " STA    49
49664 " JSR 63184   ;HEADER UMWANDELN
49667 " BIT  7168   ;WARTEN AUF SYNC
49670 " BMI 49667
49672 " LDA  7169
49675 " CLV
49676 " LDY    #0
49678 " BVC 49678
49680 " CLV
49681 " LDA  7169
49684 " CMP   768,Y ;DEN RICHTIGEN HEADER SUCHEN
49687 " BNE 49667
49689 " INY
49690 " CPY    #5
49692 " BNE 49678
49694 " LDY    #0
49696 " BIT  7168   ;WARTEN AUF SYNC
49699 " BMI 49696
49701 " LDA  7169
49704 " CLV
49705 " BVC 49705
49707 " CLV
```


Listing 2.
Der Source-Code zu »Track40«. Bitte nicht eingeben, dient nur zur Dokumentation

```
49708 " LDA  7169
49711 " STA   768,Y ;ERSTEN TEIL DES BLOCKES
49714 " INY
49715 " BNE 49705
49717 " LDY  #186
49719 " BVC 49719
49721 " CLV
49722 " LDA  7169
49725 " STA   256,Y ;UND DANN DEN ZWEITEN TEIL LESEN UND SPEICHERN
49728 " INY
49729 " BNE 49719
49731 " JSR 63712   ;IN HEX-BYTES UMWANDELN
49734 " LDA    56
49736 " CMP   #13   ;SCHAUEN OB DATENBLOCK-KENNZEICHEN STIMMT
49738 " BNE 49638   ;WENN NICHT ALLES VON VORNE
49740 " LDA    #0
49742 " STA    48
49744 " LDA    #3
49746 " STA    49   ;ZEIGER WIEDER AUF $300
49748 " JSR 62953   ;PRUEFSUMME BERECHNEN
49751 " CMP    58   ;KONTROLLE
49753 " BNE 49638   ;WENN FALSCH NOCHMALS
49755 " LDA    #0
49757 " TAY
49758 " CLC
49759 " ADC   768,Y ;INTERNE PRUEFSUMME BERECHNEN
49762 " INY
49763 " CPY  #255
49765 " BNE 49758
49767 " CMP  1023   ;UND KONTROLLE
49770 " BNE 49753
49772 " JMP 64926   ;ZURUECK ZUM IRQ
49775 " LDA  2046
49778 " STA    10
49780 " LDA  #224
49782 " STA     2
49784 " LDA     2
49786 " BMI 49784
49788 " RTS         ;ENDE
49789 ".BY  'I' '#2' 'B-P 2 0' 8  ;DATEN (DIE >8< IST DIE DRIVENUMMER)
49800 " LDA  #226   ;ROUTINE ZUM SENDEN EINER DER DREI FLOPPY-PRG. (FORMATIEREN)
49802 ".BY    44    ;BIT ...
49803 " LDA  #252   ;EINSPRUNG (SCHREIBEN)
49805 ".BY    44    ;BIT ...
49806 " LDA  #159   ;EINSPRUNG (LESEN)
49808 " PHA         ;RETTEN
49809 " JSR 65511   ;CLALL ALLE KANAELE SCHLIESSEN
49812 " LDA    #1   ;LOG. FILE NR.
49814 " LDX 49799   ;GERAETE-ADRESSE (8)
49817 " LDY   #15   ;SEKUNDAERADR.
49819 " JSR 65466   ;SETLFS
49822 " LDA    #1   ;LAENGE
49824 " LDX  #125   ;ADRESSE
49826 " LDY  #194
49828 " JSR 65469   ;SETNAM
49831 " JSR 65472   ;OPEN (DAS GANZE ENTSPRICHT OPEN1,8,15,"I")
49834 " BCC 49839   ;WENN KEIN FEHLER
49836 " JMP 57593   ;FEHLERAUSWERTUNG
49839 " LDA    #2
49841 " LDX 49799
49844 " TAY
49845 " JSR 65466   ;SETLFS
49848 " LDA    #2
49850 " LDX  #126
49852 " LDY  #194
49854 " JSR 65469   ;SETNAM
49857 " JSR 65472   ;OPEN (OPEN2,8,2,"#2")
49860 " BCS 49836   ;FEHLER?
49862 " JSR 65484   ;CLRCHN (ALLE AUSGABE- EINGABEKANAELE RUECKSETZEN)
49865 " LDX    #2
49867 " JSR 65478   ;CHKIN (KANAL 2 AUF EINGABE RICHTEN)
49870 " JSR 65487   ;CHRIN (EIN BYTE VOM SERIELLEN BUS HOLEN-PUFFERNUMMER)
49873 " JSR 65484   ;CHRCHN
49876 " LDX    #1
49878 " JSR 65481   ;CHKOUT (KANAL 1 ALS AUSGABE DEFINIEREN)
49881 " BCS 49836   ;FEHLER?
49883 " LDY    #0
```

```
49885 " LDA 49792,Y ;'B-P 2 0'
49888 " JSR 65490   ;CHROUT  (EIN BYTE AUSGEBEN)
49891 " INY
49892 " CPY     #7  ;SCHAUEN OB ALLES
49894 " BNE 49885
49896 " JSR 65484   ;CLRCHN
49899 " LDX     #2
49901 " JSR 65481   ;CHKOUT
49904 " BCS 49836
49906 " PLA         ;DIE AM ANFANG GERETTE LAENGE DES PRG. WIEDER HOLEN
49907 " STA    143  ;UND ABSPEICHERN
49909 " LDY     #0
49911 " LDA    143  ;SCHAUEN WELCHES
49913 " CMP   #226
49915 " BNE 49923
49917 " LDA 49152,Y ;FORMATIEREN
49920 " JMP 49940
49923 " CMP   #252
49925 " BNE 49933
49927 " LDA 49378,Y ;SCHREIBEN
49930 " JMP 49940
49933 " CMP   #159
49935 " BNE 49836
49937 " LDA 49630,Y ;LESEN
49940 " JSR 65490   ;GESENDET WERDEN MUSS (CHROUT)
49943 " BCS 49836   ;FEHLER?
49945 " INY
49946 " CPY    143  ;SCHAUEN OB DAS GANZE PRG. GESENDET WURDE
49948 " BNE 49911   ;WENN NICHT NAECHSTES BYTE
49950 " JSR 65484   ;CLRCHN
49953 " LDA     #2
49955 " JMP 65475   ;CLOSE (KANAL 2 WIRD GESCHLOSSEN)
49958 ".BY 'M-E' 0 5
49963 ".BY 'M-W' 254 7 2 36 0
49971 ".BY '#0' '#1'
49975 " STA 49961   ;EINSPUNGPUNKT ABSPEICHERN UND M-E SENDEN
49978 " JSR 65484   ;CLRCHN
49981 " LDX     #1
49983 " JSR 65481   ;CHKOUT
49986 " LDY     #0
49988 " LDA 49958,Y ;'M-E' X 5
49991 " JSR 65490   ;CHROUT
49994 " INY
49995 " CPY     #5
49997 " BNE 49988
49999 " RTS
50000 " LDA   #102  ;**STARTADRESSE**
50002 " STA    776
50005 " LDA   #195
50007 " STA    777  ;BASIC-BEFEHL ABARBEITUNGS-SCHLAUFE AUF DIE EIGENE ROUTINE
50010 " RTS         ;BIEGEN
50011 " LDA   #228  ;**ENDADRESSE**
50013 " LDY   #167
50015 " STA    776
50018 " STY    777  ;INTERPRETER-ZEIGER WIEDER RICHTEN,UM DIESE ERWEITERUNG
50021 " RTS         ;AUSZUSCHALTEN
50022 " JSR    115  ;**EINSPUNGPUNKT** VOM INTERPRETER HER (EIN PRG.BYTE HOLEN)
50025 " CMP    #91  ;SCHAUEN OB ES DAS '['-ZEICHEN IST
50027 " BEQ 50032   ;WENN JA DANN WEITER
50029 " JMP 50644   ;WENN NICHT DANN ZUR ORIGINAL-ROUTINE
50032 " JSR    115  ;NAECHSTES BASIC-BYTE HOLEN
50035 " CMP   #162  ;SCHAUEN OB ES DER 'NEW'-CODE IST
50037 " BNE 50084   ;WENN NICHT DANN WEITER-KONTROLLIEREN
50039 " JSR    115  ;NAECHSTES ZEICHEN
50042 " CMP    #93  ;UND SCHAUEN OB ']'
50044 " BEQ 50049
50046 " JMP 44808   ;WENN NEIN SYNTAX-ERROR
50049 " JSR 49800   ;FORMATIER-PRG. SENDEN
50052 " LDA   #193
50054 " JSR 49975   ;UND STARTEN
50057 " JSR 65484   ;CLRCHN (WARTEN DAS DIE ROUTINE FERTIG IST)
50060 " LDX     #1
50062 " JSR 65481   ;CHKOUT
50065 " LDA    #73  ;UND EIN 'I' SENDEN (INITIALISIERUNG DER FLOPPY)
50067 " JSR 65490   ;CHROUT
50070 " JSR 65484   ;CLRCHN
50073 " LDA     #1
```

Listing 2. Der Source-Code zu »Track40«

```
50075 " JSR 65475   ;CLOSE
50078 " JSR   115   ;NAECHSTES BYTE HOLEN
50081 " JMP 42926   ;UND ZURUECK ZUR ORIGINALROUTINE
50084 " CMP  #148   ;SCHAUEN OB ES DER 'SAVE'-CODE IST
50086 " BEQ 50091
50088 " JMP 50455   ;WENN NICHT WEITERKONTROLLIEREN
50091 " JSR   115   ;NAECHSTES ZEICHEN NACH DEM 'SAVE' HOLEN
50094 " JSR 44426   ;DIE VARIABLE (ZAHL) AUSWERTEN
50097 " JSR 47095   ;IN ZWEIBYTE-FORM WANDELN
50100 " STY   251   ;UND ABSPEICHERN
50102 " STA   252
50104 " JSR 44797   ;PRUEFEN AUF KOMMA
50107 " JSR 44426
50110 " JSR 47095   ;ZWEIBYTE-ZAHL HOLEN
50113 " STY   253
50115 " STA   254   ;UND ALS ENDADRESSE SPEICHERN
50117 " JSR 44797   ;PRUEFEN AUF KOMMA
50120 " LDA   253
50122 " SEC
50123 " SBC   251
50125 " STA   139    ;ENDADRESSE-ANFANGSADRESSE IN (139)
50127 " LDA   254
50129 " SBC   252
50131 " STA   140   ;ABSPEICHERN
50133 " BCS 50140
50135 " LDX   #20   ;WENN ENDADRESSE < ANFANGSADRESSE DANN DIVISION BY ZERO
50137 " JMP 42042   ;ERROR-AUSGABE
50140 " LDX    #0
50142 " LDA   139
50144 " SEC
50145 " SBC  #254
50147 " STA   139
50149 " LDA   140
50151 " SBC    #0
50153 " STA   140   ;DIFFERENZ DURCH 254 DIVIDIEREN (BLOCKLAENGE)
50155 " BCC 50160   ;WENN FERTIG
50157 " INX
50158 " BNE 50142   ;WEITER DIVIDIEREN
50160 " CPX   #79   ;SCHAUEN OB DAS FILE LAENGER ALS 78 BLOECKE WIRD
50162 " BCC 50169
50164 " LDX   #15   ;WENN JA DANN OVERFLOW ERROR
50166 " JMP 42042
50169 " TXA
50170 " PHA         ;BLOCKANZAHL RETTEN
50171 " JSR 47006   ;DEN ANFANGSBLOCK VOM BASIC INS X-REGISTER HOLEN
50174 " TXA
50175 " PHA         ;UND RETTEN
50176 " JSR   121   ;EIN BASIC-ZEICHEN HOLEN
50179 " CMP   #93   ;UND SCHAUEN OB ']'
50181 " BEQ 50186
50183 " JMP 44808   ;WENN NEIN DANN SYNTAX ERROR
50186 " JSR   115   ;NAECHSTES ZEICHEN HOLEN (VORBEREITUNG FUER DEN RUECK-
50189 " PLA         ;-SPRUNG ZUM INTERPRETER) ANFANGSBLOCK WIEDER HOLEN
50190 " STA   139   ;UND ABSPEICHERN
50192 " PLA         ;FILE-LAENGE HOLEN
50193 " CLC
50194 " ADC   139   ;ANFANG DAZUADDIEREN
50196 " CMP   #79   ;UND SCHAUEN OB DER LETZTE BLOCK > 78 IST
50198 " BCC 50205
50200 " LDX   #14   ;WENN JA DANN ILLEGAL QUANTITY ERROR
50202 " JMP 42042   ;AUSGEBEN
50205 " JSR 49803   ;DAS FLOPPY-PROGRAMM ZUM SCHREIBEN SENDEN
50208 " LDA    #2
50210 " LDX 49799
50213 " TAY
50214 " JSR 65466   ;SETLFS
50217 " LDA    #2
50219 " LDX   #53
50221 " LDY  #195
50223 " JSR 65469   ;SETNAM
50226 " JSR 65472   ;OPEN (OPEN2,8,2,"#1")
50229 " BCC 50305   ;WENN KEIN FEHLER DANN WEITER
50231 " JMP 57593   ;FEHLER
50234 " JSR 65484   ;CLRCHN   UNTERROUTINE ZUM SENDEN VON 'B-P 2 0'
50237 " LDX    #1
50239 " JSR 65481   ;CHKOUT
50242 " LDY    #0
```

```
50244 " LDA 49792,Y   ;'B-P 2 0'
50247 " JSR 65490     ;SENDEN
50250 " BCS 50231
50252 " INY
50253 " CPY     #7
50255 " BNE 50244
50257 " JMP 65484     ;CLRCHN
50260 " LDA    139    ;ROUTINE ZUM SENDEN VON SPUR UND SEKTOR('M-W' 254 7 2 X Y)
50262 " LSR           ;BLOCK LADEN
50263 " LSR           ;UND DURCH 16 TEILEN
50264 " LSR
50265 " LSR
50266 " CLC
50267 " ADC    #36
50269 " STA 49969     ;+36=AKTUELLE SPUR
50272 " LDA    139
50274 " AND    #15
50276 " STA 49970     ;BLOCK AND 15 = AKTUELLER SEKTOR
50279 " JSR 65484     ;CLRCHN
50282 " LDX     #1
50284 " JSR 65481     ;CHKOUT
50287 " LDY     #0
50289 " LDA 49963,Y   ;'M-W' 254 7 2 X Y
50292 " JSR 65490     ;SENDEN
50295 " BCS 50231
50297 " INY
50298 " CPY     #8
50300 " BNE 50289
50302 " JMP 65484     ;CLRCHN UND ENDE UNTERROUTINE
50305 " JSR 65484     ;CLRCHN
50308 " LDX     #2
50310 " JSR 65478     ;CHKIN
50313 " JSR 65487     ;CHRIN
50316 " JSR 65463     ;STATUS
50319 " BEQ 50324
50321 " JMP 42082     ;WENN <>0 DANN 'ERROR'
50324 " JSR 50260     ;AKTUELLEN TRACK UND SEKTOR SENDEN
50327 " JSR 50234     ;'B-P 2 0' SENDEN
50330 " LDX     #2
50332 " JSR 65481     ;CHROUT
50335 " LDA    253
50337 " SEC
50338 " SBC    251
50340 " STA    140
50342 " LDA    254
50344 " SBC    252    ;SCHAUEN OB DAS DER LETZTE BOLCK DES FILES IST
50346 " BNE 50354     ;WENN NEIN
50348 " LDY    140
50350 " CPY   #254
50352 " BCC 50399     ;WENN JA
50354 " LDA   #255  ;WENN ETWAS ANDERES ALS 255 GESENDET WIRD SO IST DAS DAS
50356 " JSR 65490     ;KENNZEICHEN DAS DER BETREFFENDE BLOCK DER LETZTE IST.
50359 " LDY     #0
50361 " LDA   (251),Y
50363 " JSR 65490     ;CHROUT 254 DATENBYTES UEBERTRAGEN
50366 " BCS 50295
50368 " INY
50369 " CPY   #254
50371 " BNE 50361
50373 " JSR 65484     ;CLRCHN
50376 " LDA   #222
50378 " JSR 49975     ;FLOPPY-ROUTINE STARTEN
50381 " NOP
50382 " NOP
50383 " INC    139    ;ZEIGER AUF NAECHSTEN BLOCK
50385 " LDA    251
50387 " CLC
50388 " ADC   #254
50390 " STA    251    ;ANFANGSADRESSE ERHOEHEN
50392 " BCC 50396
50394 " INC    252
50396 " JMP 50324     ;NAECHSTEN BLOCK SENDEN
50399 " INY           ;ADRESSE DES LETZTEN GULTIGEN BYTES SENDEN
50400 " TYA
50401 " JSR 65490     ;CHROUT
50404 " STY    141    ;ADRESSE RETTEN
50406 " LDY     #0
```

Listing 2.
Der Source-Code zu »Track40«

```
50408 " LDA  (251),Y
50410 " JSR 65490    ;CHROUT GUELTIGEN TEIL DES BLOCKES SENDEN
50413 " INY
50414 " CPY   141
50416 " BNE 50408
50418 " JSR 65484    ;CLRCHN
50421 " LDA  #222
50423 " JSR 49975    ;FLOPPY-ROUTINE STARTEN
50426 " JSR 65484    ;CLRCHN
50429 " LDA    #2
50431 " JSR 65475    ;CLOSE2
50434 " LDX    #1
50436 " JSR 65481
50439 " LDA   #73
50441 " JSR 65490    ;PRINT#1,"I"
50444 " JSR 65484
50447 " LDA    #1
50449 " JSR 65475    ;CLOSE1
50452 " JMP 42926    ;RETOUR ZUR INTERPRETER-SCHLEIFE
50455 " CMP  #147    ;SCHAUEN OB ES DER 'LOAD'-CODE IST
50457 " BEQ 50462
50459 " JMP 50650    ;NAECHSTER BEFEHL
50462 " JSR   115
50465 " JSR 44426
50468 " JSR 47095
50471 " STY   251
50473 " STA   252    ;ANFANGSADRESE HOLEN
50475 " JSR 44797
50478 " JSR 47006    ;ANFANGSBLOCK HOLEN
50481 " CPX   #79    ;WENN BLOCKNR. >78
50483 " BCC 50490
50485 " LDX   #14    ;DANN ILLEGAL QUANTITY ERROR AUSGEBEN
50487 " JMP 42042
50490 " STX   139    ;ANFANGSBLOCK RETTEN
50492 " JSR   121
50495 " JSR 50703    ;KONTROLLE AUF ']'
50498 " NOP
50499 " JSR   115
50502 " JSR 49806    ;FLOPPY-PRG. ZUM LESEN SENDEN
50505 " LDA    #2
50507 " LDX 49799
50510 " TAY
50511 " JSR 65466    ;SETLFS
50514 " LDA    #2
50516 " LDX   #51
50518 " LDY  #195
50520 " JSR 65469    ;SETNAM
50523 " JSR 65472    ;OPEN (OPEN2,8,2,"#0")
50526 " BCC 50531
50528 " JMP 57593    ;FEHLER
50531 " JSR 65484    ;CLRCHN
50534 " LDX    #2
50536 " JSR 65478    ;CHKIN
50539 " JSR 65487    ;CHRIN
50542 " JSR 65463    ;STATUS
50545 " BEQ 50550
50547 " JMP 42082    ;WENN ST <> 0 DANN 'ERROR'
50550 " JSR 50260    ;SPUR UND SEKTOR SENDEN
50553 " LDA  #145
50555 " JSR 49975    ;BLOCK LESEN (FLOPPY-ROUTINE STARTEN
50558 " JSR 50234    ;'B-P 2 0' SENDEN
50561 " LDX    #2
50563 " JSR 65478    ;CHKIN
50566 " JSR 65487    ;CHRIN
50569 " STA   141    ;1. BYTE ALS ANZAHL GUELTIGER BYTES +1 RETTEN
50571 " LDY    #0
50573 " JSR 65487
50576 " BCS 50528    ;FEHLER?
50578 " STA  (251),Y ;ALLE BYTES HOLEN UND SPEICHERN
50580 " INY
50581 " CPY   141    ;SCHAUEN OB ALLE GUELTIGEN BYTES
50583 " BNE 50573    ;WENN NEIN NAECHSTES
50585 " JSR 65484    ;CLRCHN
50588 " LDA   141
50590 " CMP  #255    ;SCHAUEN OB ALLE BYTES GUELTIG WAREN
50592 " BNE  50618   ;WENN NEIN DANN WAR ES DER LETZTE BLOCK
50594 " LDA   251    ;WENN JA
```

```
50596 " CLC
50597 " ADC    #254
50599 " STA     251    ;ADRESSE =ADRESSE+254
50601 " BCC   50605
50603 " INC     252
50605 " INC     139    ;BLOCK-ZEIGER UM EINS ERHOEHEN
50607 " LDA     139
50609 " CMP     #79    ;WENN BLOCK-ZEIGER ZU HOCH (>78)
50611 " BCC   50550    ;WENN NICHT NAECHSTEN BLOCK HOLEN
50613 " LDX     #14    ;DANN ILLEGAL QUANTITY ERROR
50615 " JMP   42042    ;AUSGEBEN
50618 " LDA      #2
50620 " JSR   65475    ;CLOSE2
50623 " LDX      #1
50625 " JSR   65481    ;CHKOUT
50628 " LDA     #73
50630 " JSR   65490    ;PRINT#1,"I"
50633 " JSR   65484    ;CLRCHN
50636 " LDA      #1
50638 " JSR   65475    ;CLOSE1
50641 " JMP   42926    ;ZURUECK ZUR INTERPRETER-SCHLAUFE
50644 " JSR     121
50647 " JMP 42983      ;EINSPRUNG FUER BEFEHL-ABARBEITEN
50650 " CMP    #161    ;SCHAUEN OB GET
50652 " BEQ 50657
50654 " JMP 50839      ;WENN NICHT WEITER
50657 " JSR     115
50660 " JSR 45195      ;STRINGZEIGER-ADRESSE HOLEN
50663 " STA      73
50665 " STY      74    ;UND SPEICHERN
50667 " LDA    #255    ;LAENGE
50669 " JSR 46197      ;PLATZ IM STRING-BEREICH RESERVIEREN
50672 " LDY      #2
50674 " LDA    97,Y    ;NEUE ADRESSE
50677 " STA  (73),Y    ;BEIM STRINGZEIGER RICHTEN
50679 " PHA            ;UND RETTEN
50680 " DEY
50681 " BPL 50674
50683 " JSR 44797      ;PRUEFEN AUF KOMMA
50686 " JSR 47006      ;BLOCK-NR. HOLEN
50689 " CPX     #79
50691 " BCC 50698
50693 " LDX     #14    ;WENN ZU GROSS DANN ILLEGAL QUANTITY
50695 " JMP 42042      ;AUSGEBEN
50698 " STX     139    ;BLOCK SPEICHERN
50700 " JMP 50711
50703 " CMP     #93    ;EINSCHUB VON EINER VORHERIGEN ROUTINE
50705 " BEQ 50710      ;(PRUEFEN AUF ']')
50707 " JMP 44808      ;SYNTAX-ERROR
50710 " RTS
50711 " JSR     121
50714 " JSR 50703      ;PRUEFEN AUF ']'
50717 " JSR     115    ;ZAEHLER AUF NAECHSTEN BEFEHL RICHTEN
50720 " JSR 49806      ;FLOPPY-ROUTINE 'LESEN' SENDEN
50723 " PLA            ;ZEIGER UND LAENGE DES NEUEN STRINGS HOLEN
50724 " STA     141
50726 " PLA
50727 " STA     251
50729 " PLA
50730 " STA     252    ;UND SPEICHERN
50732 " LDA      #2
50734 " LDX 49799
50737 " TAY
50738 " JSR 65466      ;SETLFS
50741 " LDA      #2
50743 " LDX     #51
50745 " LDY    #195
50747 " JSR 65469      ;SETNAM
50750 " JSR 65472      ;OPEN 2,8,2,"#0"
50753 " BCC 50758
50755 " JMP 57593      ;FEHLER
50758 " JSR 65484      ;CLRCHN
50761 " LDX      #2
50763 " JSR 65478      ;CHKIN
50766 " JSR 65487      ;CHRIN
50769 " JSR 65463      ;STATUS
```


Listing 2.
Der Source-Code zu »Track40«

```
50772 " BEQ 50777      ;KONTROLLIEREN
50774 " JMP 42082      ;'ERROR' AUSGEBEN
50777 " JSR 50260      ;SPUR + SEKTOR SENDEN
50780 " LDA  #145
50782 " JSR 49975      ;ROUTINE STARTEN
50785 " JSR 50234      ;B-P 2 0
50788 " LDX    #2
50790 " JSR 65478      ;CHKIN
50793 " NOP
50794 " NOP
50795 " NOP
50796 " LDY    #0
50798 " JSR 65487      ;255 BYTES HOLEN
50801 " BCS 50755      ;FEHLER ?
50803 " STA  (251),Y   ;UND ALS STRING SPEICHERN
50805 " INY
50806 " CPY   141
50808 " BNE 50798
50810 " JSR 65484      ;CLRCHN
50813 " LDA    #2
50815 " JSR 65475      ;CLOSE2
50818 " LDX    #1
50820 " JSR 65481      ;CHKOUT
50823 " LDA   #73
50825 " JSR 65490      ;PRINT#1,"I"
50828 " JSR 65484      ;CLRCHN
50831 " LDA    #1
50833 " JSR 65475      ;CLOSE1
50836 " JMP 42926      ;ZUM INTERPRETER
50839 " CMP  #153      ;SCHAUEN OB PRINT-CODE
50841 " BEQ 50846
50843 " JMP 44808      ;WENN NEIN SYNTAX-ERROR
50846 " JSR   115
50849 " JSR 45195      ;ADRESSE DES STRING-ZEIGERS HOLEN
50852 " PHA
50853 " TYA
50854 " PHA            ;UND RETTEN
50855 " JSR 44797      ;KOMMA
50858 " JSR 47006      ;BLOCKNR. HOLEN
50861 " CPX   #79
50863 " BCC 50870      ;PRUEFEN
50865 " LDX   #14
50867 " JMP 42042      ;ILLEG. QUANTITY
50870 " STX   139      ;UND SPEICHERN
50872 " JSR   121
50875 " JSR 50703      ;PRUEFEN AUF ']'
50878 " JSR   115
50881 " PLA
50882 " STA    98
50884 " PLA
50885 " JSR 51008      ;LAENGE PRUEFEN
50888 " INY
50889 " LDA   (97),Y
50891 " STA   251
50893 " INY
50894 " LDA   (97),Y   ;ADRESSE DES EIGENTLICHEN STRINGS VOM STRING-ZEIGER HER
50896 " STA   252      ;HOLEN UND SPEICHERN
50898 " JSR 49803      ;'SCHREIBEN' SENDEN
50901 " LDA    #2
50903 " LDX 49799
50906 " TAY
50907 " JSR 65466
50910 " LDA    #2
50912 " LDX   #53
50914 " LDY  #195
50916 " JSR 65469
50919 " JSR 65472      ;OPEN2,8,2,"#1"
50922 " BCC 50927
50924 " JMP 57593      ;FEHLER
50927 " JSR 65484
50930 " LDX    #2
50932 " JSR 65478
50935 " JSR 65487
50938 " JSR 65463
50941 " BEQ 50946
50943 " JMP 42082
```

```
50946 " JSR 50260      ;SPUR + SEKTOR SENDEN
50949 " JSR 50234      ;B-P 2 0 SENDEN
50952 " LDX    #2
50954 " JSR 65481      ;CHKOUT
50957 " LDY    #0
50959 " LDA  (251),Y   ;255 DATEN AUS DEM STRING LESEN
50961 " JSR 65490      ;UND SENDEN
50964 " BCS 50924
50966 " INY
50967 " CPY  #255
50969 " BNE 50959
50971 " JSR 65484      ;CLRCHN
50974 " LDA  #222
50976 " JSR 49975      ;FLOPPY-ROUTINE STARTEN
50979 " JSR 65484
50982 " LDA    #2
50984 " JSR 65475      ;CLOSE2
50987 " LDX    #1
50989 " JSR 65481
50992 " LDA   #73
50994 " JSR 65490      ;PRINT#1,"I"
50997 " JSR 65484
51000 " LDA    #1
51002 " JSR 65475      ;CLOSE1
51005 " JMP 42926      ;ZUM INTERPRET
51008 " STA    97      ;EINSCHUB (LAENGE BESTIMMEN)
51010 " LDY    #0
51012 " LDA   (97),Y    ;LAENGE HOLEN
51014 " CMP  #255
51016 " BEQ 51023
51018 " LDX   #23       ;WENN FALSCH DANN STRING TOO LONG AUSGEBEN
51020 " JMP 42042
51023 " RTS
51024 ".EN

READY.
```


Listing 2. Der Source-Code zu »Track40«. (Schluß). Bitte nicht eingeben, dient nur zur Dokumentation.

```
Name : track40.laden      c000 c2b8
-----------------------------------
c000 : a6 8b e0 4f 90 05 a2 0e   66
c008 : 86 02 60 20 e7 ff a9 01   d3
c010 : a2 08 a0 0f 20 ba ff a9   eb
c018 : 01 a2 04 a0 50 20 bd ff   7c
c020 : 20 c0 ff a9 02 a2 08 a8   7c
c028 : 20 ba ff a9 02 a2 05 a0   65
c030 : 50 20 bd ff 20 c0 ff 20   48
c038 : cc ff a2 02 20 c6 ff 20   65
c040 : cf ff 20 cc ff a2 01 20   0a
c048 : c9 ff a0 00 b9 07 50 20   8e
c050 : d2 ff c8 c0 07 d0 f5 20   7b
c058 : cc ff a2 02 20 c9 ff a0   9e
c060 : 00 b9 65 4f 20 d2 ff c8   aa
c068 : c0 9f d0 f5 20 cc ff a9   a6
c070 : 02 20 c3 ff a9 02 a2 08   b8
c078 : a8 20 ba ff a9 02 a2 0f   32
c080 : a0 50 20 bd ff 20 c0 ff   0c
c088 : 20 cc ff a2 02 20 c6 ff   9f
c090 : 20 cf ff a5 8b 4a 4a 4a   15
c098 : 4a 18 69 24 8d 17 50 a5   eb
c0a0 : 8b 29 0f 8d 18 50 20 cc   53
c0a8 : ff a2 01 20 c9 ff a0 00   5c
c0b0 : b9 11 50 20 d2 ff c8 c0   dc
c0b8 : 08 d0 f5 f0 02 90 d4 20   fc
c0c0 : cc ff a9 91 20 cc ff a2   d6
c0c8 : 01 20 c9 ff a0 00 b9 19   6f
c0d0 : 50 20 d2 ff c8 c0 05 d0   2d
c0d8 : f5 20 cc ff a2 01 20 c9   57
c0e0 : ff a0 00 b9 07 50 20 d2   80
c0e8 : ff c8 c0 07 d0 f5 20 cc   33
c0f0 : ff a2 02 20 c6 ff 20 cf   51
c0f8 : ff 85 8d a0 00 20 cf ff   71
c100 : 91 fb c8 c4 8d d0 f6 20   d5
c108 : cc ff a5 8d c9 ff d0 18   ff
c110 : a5 fb 18 69 fe 85 fb 90   13
c118 : 02 e6 fc e6 8b a5 8b c9   51
c120 : 4f 90 9a a2 0e 86 02 60   90
c128 : a9 02 20 c3 ff a2 01 20   ac
c130 : c9 ff a9 49 20 d2 ff 20   65
c138 : cc ff a9 01 20 c3 ff a2   f4
c140 : 00 86 02 60 00 ad 00 1c   b6
c148 : 29 9f 8d 00 1c a9 0e 85   f7
c150 : 52 ad fe 07 85 53 ad ff   c3
c158 : 07 85 54 a9 00 85 55 85   f9
c160 : 34 85 30 a9 03 85 31 20   f9
c168 : d0 f6 2c 00 1c 30 fb ad   4d
c170 : 01 1c b8 a0 00 50 fe b8   b1
c178 : ad 01 1c d9 00 03 d0 ea   19
c180 : c8 c0 05 d0 f0 a0 00 2c   70
c188 : 00 1c 30 fb ad 01 1c b8   e7
c190 : 50 fe b8 ad 01 1c 99 00   9a
c198 : 03 c8 d0 f4 a0 ba 50 fe   f1
c1a0 : b8 ad 01 1c 99 00 01 c8   22
c1a8 : d0 f4 20 e0 f8 a5 38 c9   48
c1b0 : 0d d0 9a a9 00 85 30 a9   41
c1b8 : 03 85 31 20 e9 f5 c5 3a   a8
c1c0 : d0 8b a9 00 a8 18 79 00   f1
c1c8 : 03 c8 c0 ff d0 f7 cd ff   63
c1d0 : 03 d0 ed 4c 9e fd ad fe   cf
c1d8 : 07 85 0a a9 e0 85 02 a5   e7
c1e0 : 02 30 fc 60 49 23 32 42   40
c1e8 : 2d 50 20 32 20 30 20 23   d6
c1f0 : 30 4d 2d 57 fe 07 02 24   76
c1f8 : 00 4d 2d 45 91 05 a0 62   1b
c200 : a5 fd 18 69 45 85 8c 91   ad
c208 : fd a5 fe 69 01 85 8d c8   c9
c210 : 91 fd a5 8c 18 69 9f 85   f1
c218 : 8c a0 1a 91 fd a5 8d 69   c3
c220 : 00 85 8d a0 1c 91 fd a0   e2
c228 : 2e e6 8c d0 02 e6 8d a5   df
c230 : 8c 91 fd a5 8d a0 30 91   7b
c238 : fd a5 8c 18 69 02 85 8c   04
c240 : a0 4d 91 fd a0 e4 91 fd   1e
c248 : a5 8d 18 69 00 85 8d c8   db
c250 : 91 fd a0 4e 91 fd a5 8c   8a
c258 : 18 69 08 85 8c a0 7f 91   c6
c260 : fd a0 81 a5 8d 18 69 00   02
c268 : 85 8d 91 fd a0 b1 a5 8c   1f
c270 : 18 69 02 85 8c 91 fd c8   4d
c278 : a5 8d 69 00 85 8d 91 fd   45
c280 : a0 9d a5 8c 18 69 06 85   da
c288 : 8c 91 fd c8 a5 8d 69 00   e2
c290 : 85 8d 91 fd e6 8c d0 02   1a
c298 : e6 8d a0 a4 a5 8c 91 fd   02
c2a0 : a5 8d c8 91 fd e6 8c d0   5b
c2a8 : 02 e6 8d a0 cf a5 8c 91   14
c2b0 : fd a5 8d c8 91 fd 6c fd   b3
```

Listing 4. Zum Einbau in eigene Programme hier die Laderoutine. Bitte beachten Sie die Eingabehinweise auf Seite 100.

Kopieren mit zwei Laufwerken

Dieses Programm ist für alle Besitzer von zwei 1541-Floppy-Laufwerken gedacht, die einzelne Programme von Laufwerk 8 auf Laufwerk 9 übertragen wollen.

Das Programm »Dual-Filecopy« ist in der Lage, Programm (PRG)-, User (USR)- und sequentielle (SEQ) Dateien von einer 1541-Floppy (Geräteadresse 8) auf eine zweite 1541 (Geräteadresse 9) zu kopieren. Die Files können dabei mit Hilfe einer »Ja/Nein«-Abfrage aus dem Directory ausgewählt werden. Weiterhin stehen eine SCRATCH-Funktion sowie die üblichen Laufwerksfunktionen zur Verfügung.

Das gesamte Programm ist menügesteuert und vollständig in Maschinensprache programmiert.

Eingabehinweise

Das Programm (Listing 1) muß mit dem MSE eingegeben und gespeichert werden. Nach dem Laden wird es einfach mit RUN gestartet.

Vor dem Start

Will man mit dem Programm arbeiten, müssen vorher zwei Diskettenlaufwerke (VC1541 oder kompatible) an den C64 angeschlossen werden. Dabei muß sich das Quell-Laufwerk durch die Geräteadresse 8 und das Ziellaufwerk durch die Geräteadresse 9 angesprochen fühlen. Wie Sie die Geräteadressen ändern, ist im Floppy-Handbuch beschrieben. Ferner befindet sich auf der Test/Demo-Diskette ein entsprechendes Programm zur softwaremäßigen Änderung.

Bedienungsanleitung

Nach dem Start sollte sich das Programm mit einer Einschaltmeldung und einem Menü melden. Erscheint am unteren Bildschirmrand eine Fehlermeldung, so lassen sich keine Menüpunkte anwählen, da ein Fehler beim Testen der angeschlossenen Laufwerke vorliegt. Entweder ist nur eines angeschlossen, oder die Geräteadressen stimmen nicht. Bringen Sie bitte zuerst Ihre Hardwarekonfiguration in Ordnung und starten dann das Programm erneut.

Die Menüpunkte lassen sich über die Funktionstasten anwählen. Es stehen die Befehle COPY, SCRATCH, ORDER (Befehl) #8, ORDER (Befehl) #9, STATUS #8, STATUS #9, DIRECTORY #8 und DIRECTORY #9 zur Verfügung. Die <RUN/STOP>-Taste dient jeweils an bestimmten Programmstellen zum Unterprogrammabbruch.

Die Befehle, die die Funktionstasten <F3> bis <F7> belegen, beziehen sich immer auf Laufwerk #8, die geSHIFTeten Funktionstasten (F4 bis F8) sprechen Laufwerk #9 an (zum Beispiel: <F7> Directory #8, <F8> Directory #9).

Die Funktionstasten sind folgendermaßen belegt:

<F1> - Copy Files #8 to #9

Files (Dateien) werden grundsätzlich von Laufwerk #8 auf Laufwerk #9 kopiert. Man wird aufgefordert, die Disketten einzulegen und die zu kopierenden Dateien auszusuchen. Es werden nur die File-Typen »PRG«, »SEQ« und »USR« angenommen. Andere Files kann das Programm nicht kopieren und werden deshalb übersprungen. Nach einer Sicherheitsabfrage ist zwischen »Bildschirm eingeschaltet« und »Bildschirm ausgeblendet« zu wählen. Ein Ausblenden des Bildschirms hat allerdings keinen Einfluß auf die Geschwindigkeit.

Anschließend wird File für File angezeigt, kopiert und der Fehlerkanal ausgelesen. Das Programm meldet sich wieder mit »COPY COMPLETE !« und der Kopiervorgang ist beendet.

<F2> - Scratch Files #8

Es können beliebige Dateien auf Laufwerk #8 gelöscht werden. Das Programm fordert Sie auf, die Diskette mit den zu löschenden Files in Laufwerk #8 einzulegen und eine Taste zu drücken. Nun können die zu löschenden Dateien aus dem Directory angewählt werden.

<RUN/STOP> bricht den Vorgang ab und verzweigt wieder ins Hauptmenü. <RETURN> beendet das Selektieren der Files.

Alle folgenden Dateien werden als nicht selektiert behandelt. Nach einer Sicherheitsabfrage beginnt das Programm die ausgewählten Dateien anzuzeigen und zu löschen.

<F3> - Disk Error #8

Anzeigen des Fehlerkanals von Laufwerk #8

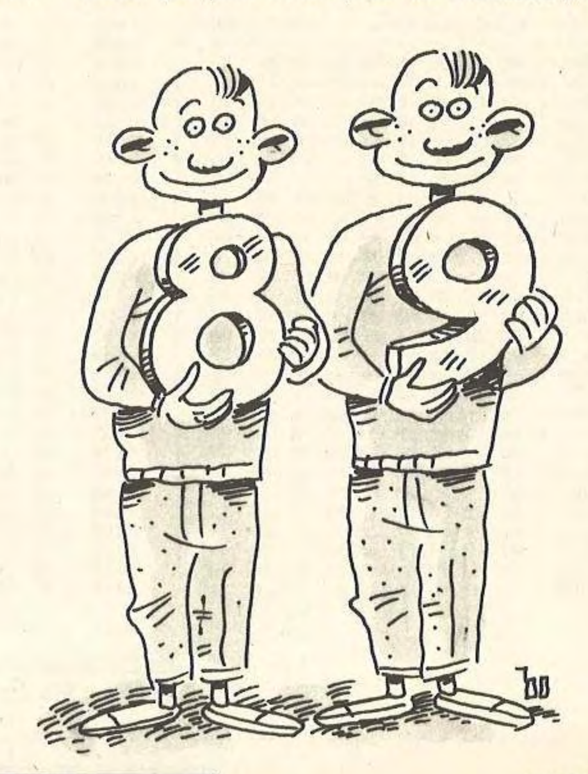
<F4> - Disk Error #9

Anzeigen des Fehlerkanals von Laufwerk #9

<F5> - Order #8

Hier können die üblichen DOS-Befehle an das Laufwerk übergeben werden (VALIDATE, NEW...). Die Eingabe des Befehls erfolgt über einen »INPUT« im unteren Bildschirmbereich. Anschließend wird der Fehlerkanal ausgelesen und angezeigt. Ein Tastendruck führt Sie zurück ins Hauptmenü.

<F6> - Order #9 (wie <F5>, nur für Laufwerk #9).



<F7> - Directory #8

Es wird das Directory der Diskette in Laufwerk #8 angezeigt. Die Ausgabe kann mit <RUN/STOP> abgebrochen werden.

<F8> - Directory #9 (wie <F7>, nur für Laufwerk #9).

Das Dual-Filecopy zählt zwar nicht zu den schnellsten Kopierprogrammen, doch wer zwei Laufwerke besitzt, kann sich diesen etwas höheren Zeitaufwand leisten, da er nicht ständig neben dem Computer sitzen muß, um auf den nächsten Diskettenwechsel zu warten. In puncto Sicherheit und Komfort steht das Dual-Filecopy professionellen Kopierprogrammen nicht nach. Kopierprogramme für zwei Laufwerke sind ohnehin kaum im Handel zu haben.

(Thomas Hohenberger/dm)

```
Name : dual-filecopy          0801 1349
-----------------------------------------
0801 : 34 08 00 00 9e 32 31 30   da
0809 : 34 3a 22 14 14 14 14 14   c0
0811 : 14 14 14 14 28 43 29 20   38
0819 : 31 39 38 36 20 42 59 20   75
0821 : 54 48 4f 4d 41 53 20 48   d7
0829 : 4f 48 45 4e 42 45 52 47   de
0831 : 45 52 00 00 00 00 00 20   e0
0839 : e7 ff a9 0c a2 0f 8d 20   25
0841 : d0 8e 21 d0 a9 b9 a2 0f   cc
0849 : 20 54 0f 20 be 0e a2 08   52
0851 : 20 a1 0f 8d 31 13 bd 16   86
0859 : 13 cd 31 13 f0 06 ca 10   8c
0861 : f5 4c 4f 08 8a 0a aa bd   70
0869 : 1f 13 85 fb bd 20 13 85   27
0871 : fc 6c fb 00 68 68 4c 74   86
0879 : a4 a9 08 4c d6 0d a9 09   0c
0881 : 4c d6 0d a9 08 4c 01 0e   b4
0889 : a9 09 4c 01 0e a9 08 4c   d1
0891 : a3 0e a9 09 4c a3 0e a9   34
0899 : 00 8d 33 13 a9 fa a2 11   ae
08a1 : 20 54 0f 20 2e 0b 20 ae   cc
08a9 : 0b 20 3d 0b b0 22 20 44   9a
08b1 : 0b b0 08 ad 11 d0 29 ef   e8
08b9 : 8d 11 d0 20 d4 08 ad 11   6d
08c1 : d0 09 10 8d 11 d0 a9 03   10
08c9 : a2 13 20 54 0f 20 b3 0f   66
08d1 : 4c 38 08 a9 6a a2 13 85   84
08d9 : fd 86 fe 20 61 0b 90 01   90
08e1 : 60 a9 cd a2 12 20 54 0f   6f
08e9 : 20 94 0b a9 00 85 90 a9   0d
08f1 : 30 8d 23 20 8d 4b 20 20   a9
08f9 : d3 09 a9 03 a2 08 a0 0f   27
0901 : 20 ba ff 20 c0 ff 20 d3   b6
0909 : 09 a9 04 a2 09 a0 0f 20   4e
0911 : ba ff 20 c0 ff ac 48 13   98
0919 : a9 2c 99 34 13 c8 a9 52   88
0921 : 99 34 13 c8 8c 48 13 98   3b
0929 : a2 34 a0 13 20 bd ff a9   b3
0931 : 01 a2 08 a0 02 20 ba ff   a6
0939 : 20 c0 ff a5 90 f0 03 4c   a3
0941 : ad 09 a9 08 20 e1 09 b0   75
0949 : 63 ac 48 13 88 a9 57 99   dd
0951 : 34 13 c8 98 a2 34 a0 13   c8
0959 : 20 bd ff a9 02 a2 09 a0   27
0961 : 02 20 ba ff 20 c0 ff a9   7d
0969 : 00 8d 20 1f a5 90 f0 03   c4
0971 : 4c ad 09 a9 09 20 e1 09   37
0979 : b0 32 20 5c 0a a9 08 20   24
0981 : e1 09 b0 28 20 82 0a b0   b8
0989 : 23 a9 09 20 e1 09 b0 1c   29
0991 : ad 20 1f f0 e5 20 cc ff   c7
0999 : a9 01 20 c3 ff a9 02 20   d9
09a1 : c3 ff a9 08 20 e1 09 a9   58
09a9 : 09 20 e1 09 20 cc ff a9   18
09b1 : 08 85 02 20 2d 0a e6 02   c3
09b9 : 20 2d 0a a9 01 20 c3 ff   48
09c1 : a9 02 20 c3 ff a9 03 20   85
09c9 : c3 ff a9 04 20 c3 ff 4c   2f
09d1 : dc 08 a9 49 8d 31 13 a9   47
09d9 : 01 a2 31 a0 13 4c bd ff   16
09e1 : 85 02 a2 03 c9 08 f0 02   15
09e9 : a2 04 20 c6 ff a0 00 8c   8c
09f1 : 31 13 20 cf ff ac 31 13   fe
09f9 : 8d 22 20 a5 02 c9 08 d0   84
0a01 : 09 ad 22 20 99 23 20 4c   39
0a09 : 11 0a ad 22 20 99 4b 20   0b
0a11 : c9 0d f0 03 c8 d0 d8 20   b4
0a19 : cc ff ad 23 20 c9 30 d0   67
0a21 : 09 ad 4b 20 c9 30 d0 02   3d
0a29 : 18 60 38 60 20 70 0f a2   93
0a31 : 00 a5 02 c9 08 d0 12 bd   88
0a39 : 23 20 8e 31 13 20 d2 ff   b4
0a41 : ae 31 13 e8 c9 0d d0 ef   92
0a49 : 60 bd 4b 20 8e 31 13 20   5e
0a51 : d2 ff ae 31 13 e8 c9 0d   ae
0a59 : d0 ef 60 a2 01 20 c6 ff   b9
0a61 : a2 00 8e 21 1f 20 cf ff   fd
0a69 : ae 21 1f 9d 22 1f a5 90   f6
0a71 : d0 08 e8 e0 fe f0 06 4c   c4
0a79 : 63 0a ce 20 1f 20 cc ff   bf
0a81 : 60 a2 02 20 c9 ff a2 00   de
0a89 : bd 22 1f 8e 31 13 20 d2   c3
0a91 : ff ae 31 13 a5 90 d0 09   ca
0a99 : ec 21 1f f0 09 e8 4c 89   18
0aa1 : 0a 20 cc ff 38 60 20 cc   8f
0aa9 : ff 18 60 a9 80 8d 33 13   69
0ab1 : a9 45 a2 12 20 54 0f 20   09
0ab9 : 2e 0b 20 ae 0b 20 3d 0b   07
0ac1 : b0 03 20 c9 0a 4c 38 08   28
0ac9 : a9 6a a2 13 85 fd 86 fe   13
0ad1 : 20 61 0b 90 18 a9 08 20   a6
0ad9 : b1 ff a9 6f 20 93 ff a9   d4
0ae1 : 49 20 a8 ff a9 30 20 a8   52
0ae9 : ff 20 ae ff 60 a9 be a2   37
0af1 : 12 20 54 0f 20 94 0b a9   30
0af9 : 08 20 b1 ff a9 6f 20 93   3b
0b01 : ff a9 53 20 a8 ff a9 30   3f
0b09 : 20 a8 ff a9 3a 20 a8 ff   fa
0b11 : a9 00 8d 31 13 ac 31 13   c5
0b19 : b9 34 13 c9 2c f0 08 20   95
0b21 : a8 ff ee 31 13 d0 ee 20   5e
0b29 : ae ff 4c d1 0a 20 a1 0f   6a
0b31 : f0 fb c9 03 d0 05 68 68   99
0b39 : 4c 38 08 60 a9 d9 a2 12   c8
0b41 : 4c 48 0b a9 e8 a2 12 20   d5
0b49 : 54 0f 20 a1 0f c9 4a f0   ab
0b51 : 0b c9 4e f0 09 c9 03 f0   bf
0b59 : 05 4c 4b 0b 18 60 38 60   df
0b61 : a0 00 b1 fd d0 07 c8 b1   f9
0b69 : fd d0 02 38 60 a0 00 b1   c4
0b71 : fd 99 34 13 c9 2c f0 03   72
0b79 : c8 d0 f4 c8 b1 fd 99 34   d9
0b81 : 13 c8 8c 48 13 98 18 65   46
0b89 : fd 85 fd a5 fe 69 [illegible] [illegible]   c3
0b91 : fe 18 60 a0 00 b9 34 13   8c
0b99 : c9 2c f0 06 20 d2 ff c8   9f
0ba1 : d0 f3 a9 22 20 d2 ff a9   05
0ba9 : 0d 20 d2 ff 60 a9 30 8d   aa
0bb1 : 23 20 8d 4b 20 a9 00 85   0b
0bb9 : 90 20 cc ff a9 6a a2 13   2b
0bc1 : 85 fd 86 fe a9 03 a2 08   14
0bc9 : a0 0f 20 ba ff a9 49 8d   de
0bd1 : 31 13 a9 01 a2 31 a0 13   73
0bd9 : 20 bd ff 20 c0 ff a5 90   9f
0be1 : d0 07 a9 08 20 e1 09 90   f7
0be9 : 0c a9 4e 8d 77 02 a9 01   3f
0bf1 : 85 c6 4c 33 0d a9 01 a2   ba
0bf9 : 08 a0 00 20 ba ff a9 24   f0
0c01 : 8d 31 13 a9 01 a2 31 a0   4c
0c09 : 13 20 bd ff 20 c0 ff a5   ef
0c11 : 90 f0 03 4c 33 0d a9 08   b6
0c19 : 20 e1 09 b0 f6 a2 01 20   4b
0c21 : c6 ff a9 06 8d 31 13 20   01
0c29 : cf ff ce 31 13 d0 f8 a9   c1
0c31 : 20 20 d2 ff 20 cf ff c9   2a
0c39 : 00 f0 06 20 d2 ff 4c 35   00
0c41 : 0c a9 0d 20 d2 ff 20 cc   b0
0c49 : ff a9 08 20 e1 09 90 03   d2
0c51 : 4c 33 0d a9 0d 20 d2 ff   cc
0c59 : 20 cc ff a2 01 20 c6 ff   60
0c61 : a2 00 8e 31 13 20 cf ff   3f
0c69 : ae 31 13 9d 4a 13 e8 e0   cb
0c71 : 20 f0 06 8e 31 13 4c 66   06
0c79 : 0c 20 cc ff a9 08 20 e1   e7
0c81 : 09 08 a2 01 20 c6 ff 28   e0
0c89 : 90 03 4c 33 0d a9 20 20   f3
0c91 : d2 ff ad 4d 13 ae 4c 13   76
0c99 : 20 cd bd a9 20 20 d2 ff   93
0ca1 : a0 04 b9 4a 13 20 d2 ff   78
0ca9 : c8 c0 20 d0 f5 a0 04 c8   f9
0cb1 : c0 20 f0 7e b9 4a 13 c9   5b
0cb9 : 22 d0 f4 a2 00 c8 b9 4a   96
0cc1 : 13 c9 22 f0 06 9d 34 13   a4
0cc9 : e8 d0 f2 a9 00 9d 34 13   ef
0cd1 : 20 4e 0d 8d 49 13 c9 00   62
0cd9 : d0 03 4c 33 0d c9 01 d0   69
0ce1 : 03 4c 54 0c 20 0a 0f 08   40
0ce9 : ad 32 13 f0 19 90 13 68   c5
0cf1 : 68 68 20 cc ff a9 01 20   c0
0cf9 : c3 ff a9 03 20 c3 ff 4c   3f
0d01 : 38 08 68 4c 33 0d 28 b0   7f
0d09 : 03 4c 54 0c a0 00 b9 34   22
0d11 : 13 91 fd f0 03 c8 d0 f6   32
0d19 : a9 2c 91 fd c8 ad 49 13   42
0d21 : 91 fd c8 98 18 65 fd 85   a6
0d29 : fd a5 fe 69 00 85 fe 4c   a6
0d31 : 54 0c a9 0d 20 d2 ff a9   83
0d39 : 00 a8 91 fd c8 91 fd 20   03
0d41 : cc ff a9 01 20 c3 ff a9   0b
0d49 : 03 20 c3 ff 60 2c 33 13   a7
0d51 : 10 03 a9 58 60 a2 18 bd   4f
0d59 : 4a 13 c9 50 f0 1a c9 53   57
0d61 : f0 2d c9 55 f0 40 e8 e0   7b
0d69 : 20 d0 ec ad 4e 13 c9 42   0b
0d71 : d0 03 a9 00 60 a9 01 60   45
0d79 : e8 bd 4a 13 c9 52 d0 da   5d
0d81 : e8 bd 4a 13 c9 47 f0 04   df
0d89 : ca 4c 58 0d a9 50 60 e8   a2
0d91 : bd 4a 13 c9 45 d0 c3 e8   2d
0d99 : bd 4a 13 c9 51 f0 04 ca   bc
0da1 : 4c 58 0d a9 53 60 e8 bd   e9
0da9 : 4a 13 c9 53 d0 ac e8 bd   eb
0db1 : 4a 13 c9 52 f0 04 ca 4c   34
0db9 : 58 0d a9 55 60 a5 02 20   28
0dc1 : b4 ff a9 6f 20 96 ff 20   c4
0dc9 : a5 ff 20 d2 ff c9 0d d0   f4
0dd1 : f6 20 ab ff 60 85 02 20   3d
0dd9 : 70 0f 20 be 0d 20 b3 0f   6f
0de1 : 4c 38 08 a5 02 20 b1 ff   e8
0de9 : a9 6f 20 93 ff a2 00 bd   55
0df1 : 6a 13 c9 0d f0 06 20 a8   0a
0df9 : ff e8 d0 f3 20 ae ff 60   57
0e01 : 85 02 20 69 0f a2 00 20   03
0e09 : cf ff 9d 6a 13 e8 c9 0d   46
0e11 : d0 f5 20 e4 0d a9 0d 20   13
0e19 : d2 ff 20 d2 ff a9 20 20   5b
0e21 : d2 ff 20 be 0d 20 b3 0f   91
0e29 : 4c 38 08 a9 24 8d 31 13   62
0e31 : a9 01 a2 31 a0 13 20 bd   c8
0e39 : ff a9 01 a6 02 a0 00 20   87
0e41 : ba ff 20 c0 ff a2 01 20   74
0e49 : c6 ff a0 05 8c 31 13 20   b6
0e51 : cf ff aa ac 31 13 88 d0   cf
0e59 : f3 a9 20 20 d2 ff 20 cf   7a
0e61 : ff a4 90 d0 34 20 cd bd   e7
0e69 : a9 20 20 d2 ff 20 cf ff   c5
0e71 : 8d 31 13 c9 00 f0 09 ad   9c
0e79 : 31 13 20 d2 ff 4c 6e 0e   ce
0e81 : a9 0d 20 d2 ff 20 cc ff   47
0e89 : 20 a1 0f c9 03 f0 0a a2   9c
0e91 : 01 20 c6 ff a0 03 4c 4d   42
0e99 : 0e 20 cc ff a9 01 20 c3   95
0ea1 : ff 60 85 02 20 8e 0f 20   65
0ea9 : 2c 0e a9 0d 20 d2 ff a9   d4
0eb1 : 20 20 d2 ff 20 be 0d 20   02
0eb9 : b3 0f 4c 38 08 a9 00 85   e7
0ec1 : 90 20 bd ff a9 01 a2 09   10
0ec9 : 86 02 a0 00 20 ba ff 20   91
0ed1 : c0 ff a9 01 20 c3 ff a5   87
0ed9 : 90 d0 1f a9 00 85 90 20   7d
0ee1 : bd ff a9 01 a2 08 86 02   b1
0ee9 : a0 00 20 ba ff 20 c0 ff   ed
0ef1 : a9 01 20 c3 ff a5 90 d0   ac
0ef9 : 01 60 a9 c1 a2 11 20 54   a9
0f01 : 0f 68 68 20 b3 0f 4c 38   b8
0f09 : 08 a9 81 a2 12 20 54 0f   2c
0f11 : 20 42 f1 c9 4a f0 0f c9   04
0f19 : 4e f0 19 c9 03 f0 23 c9   37
0f21 : 0d f0 26 4c 11 0f a9 ae   47
0f29 : a2 12 20 54 0f a9 00 8d   c0
0f31 : 32 13 38 60 a9 b7 a2 12   0e
0f39 : 20 54 0f a9 00 8d 32 13   d8
0f41 : 18 60 a9 ff 8d 32 13 38   1b
0f49 : 60 20 35 0f a9 ff 8d 32   1e
0f51 : 13 18 60 85 fb 86 fc a0   62
0f59 : 00 b1 fb f0 0a 20 d2 ff   3c
0f61 : c8 d0 f6 e6 fc d0 f2 60   0f
0f69 : a9 a7 a2 11 4c 74 0f a9   a9
0f71 : ad a2 11 20 54 0f a5 02   10
0f79 : 09 30 20 d2 ff a9 20 20   0b
0f81 : d2 ff a9 3a 20 d2 ff a9   f0
```

Listing 1. »Dual-Filecopy«. Bitte mit dem MSE (Seite 100) eingeben

```
Øf89 : 2Ø 2Ø d2 ff 6Ø a9 b6 a2  e1
Øf91 : 11 2Ø 54 Øf 2Ø 7Ø Øf a9  be
Øf99 : Ød 2Ø d2 ff 2Ø d2 ff 6Ø  c4
Øfa1 : 8a 48 98 48 2Ø e4 ff 8d  c3
Øfa9 : 31 13 68 a8 68 aa ad 31  88
Øfb1 : 13 6Ø 2Ø a1 Øf fØ fb 6Ø  5a
Øfb9 : 93 97 Øe Ø8 2Ø 2Ø 2Ø 2Ø  6Ø
Øfc1 : a4 a4 a4 a4 a4 a4 a4 a4  cØ
Øfc9 : a4 a4 a4 a4 a4 a4 a4 a4  c8
Øfd1 : a4 a4 a4 a4 a4 a4 a4 a4  dØ
Øfd9 : a4 a4 a4 a4 a4 a4 a4 a4  d8
Øfe1 : Ød 2Ø 2Ø 2Ø 2Ø 12 2Ø 2Ø  5e
Øfe9 : 2Ø d4 4f 4d 27 53 2Ø c4  Ø8
Øff1 : 55 41 4c 2Ø c4 52 49 56  af
Øff9 : 45 2Ø c6 49 4c 45 2d c3  54
1ØØ1 : 4f 5Ø 59 2Ø 2Ø 2Ø 92 Ød  3a
1ØØ9 : 2Ø 2Ø 2Ø 2Ø 12 2Ø 28 c3  9Ø
1Ø11 : 29 2Ø 31 39 38 36 2Ø 42  f8
1Ø19 : 59 2Ø d4 48 4f 4d 41 53  cc
1Ø21 : 2Ø c8 4f 48 45 4e 42 45  dd
1Ø29 : 52 47 45 52 2Ø 92 Ød Ød  9f
1Ø31 : Ød 2Ø 2Ø 2Ø 2Ø 2Ø d3 45  37
1Ø39 : 4c 45 43 54 2Ø c1 43 54  49
1Ø41 : 49 56 49 54 59 2Ø 3a Ød  2c
1Ø49 : 2Ø 2Ø 2Ø 2Ø 2Ø a3 a3 a3  7a
1Ø51 : a3 a3 a3 a3 a3 a3 a3 a3  5Ø
1Ø59 : a3 a3 a3 a3 a3 a3 Ød 2Ø  f7
1Ø61 : 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 bØ  4a
1Ø69 : cØ cØ cØ cØ ae 92 Ød 2Ø  c5
1Ø71 : 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 dd  b5
1Ø79 : 2Ø c6 31 2Ø dd 92 2Ø 2Ø  8Ø
1Ø81 : c3 4f 5Ø 59 2Ø c6 49 4c  21
1Ø89 : 45 53 2Ø 23 38 2Ø 54 4f  59
1Ø91 : 2Ø 23 39 Ød 2Ø 2Ø 2Ø 2Ø  f7
1Ø99 : 2Ø 2Ø 2Ø 12 dd 2Ø c6 32  72
1Øa1 : 2Ø dd 92 2Ø 2Ø d3 43 52  ab
1Øa9 : 41 54 43 48 2Ø c6 49 4c  e4
1Øb1 : 45 53 2Ø 23 38 Ød 2Ø 2Ø  b9
1Øb9 : 2Ø 2Ø 2Ø 2Ø 2Ø 12 ab cØ  b8
1Øc1 : cØ cØ cØ b3 92 Ød 2Ø 2Ø  da
1Øc9 : 2Ø 2Ø 2Ø 2Ø 2Ø 12 dd 2Ø  5Ø
1Ød1 : c6 33 2Ø dd 92 2Ø 2Ø c4  29
1Ød9 : 49 53 4b 2Ø c5 52 52 4f  7a
1Øe1 : 52 2Ø 23 38 Ød 2Ø 2Ø 2Ø  a6
1Øe9 : 2Ø 2Ø 2Ø 2Ø 12 dd 2Ø c6  44
1Øf1 : 34 2Ø dd 92 2Ø 2Ø c4 49  a8
1Øf9 : 53 4b 2Ø c5 52 52 4f 52  4c
11Ø1 : 2Ø 23 39 Ød 2Ø 2Ø 2Ø 2Ø  67
11Ø9 : 2Ø 2Ø 2Ø 12 ab cØ cØ cØ  c9
1111 : cØ b3 92 Ød 2Ø 2Ø 2Ø 2Ø  b5
1119 : 2Ø 2Ø 2Ø 12 dd 2Ø c6 35  f8
1121 : 2Ø dd 92 2Ø 2Ø cf 52 44  2b
1129 : 45 52 2Ø 23 38 Ød 2Ø 2Ø  b1
1131 : 2Ø 2Ø 2Ø 2Ø 2Ø 12 dd 2Ø  b8
1139 : c6 36 2Ø dd 92 2Ø 2Ø cf  28
1141 : 52 44 45 52 2Ø 23 39 Ød  6b
1149 : 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12  2d
1151 : ab cØ cØ cØ cØ b3 92 Ød  b2
1159 : 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12  3d
1161 : dd 2Ø c6 37 2Ø dd 92 2Ø  62
1169 : 2Ø c4 49 52 45 43 54 4f  e6
1171 : 52 59 2Ø 23 38 Ød 2Ø 2Ø  89
1179 : 2Ø 2Ø 2Ø 2Ø 2Ø 12 dd 2Ø  ØØ
1181 : c6 38 2Ø dd 92 2Ø 2Ø c4  5b
1189 : 49 52 45 43 54 4f 52 59  71
1191 : 2Ø 23 39 Ød 2Ø 2Ø 2Ø 2Ø  f7
1199 : 2Ø 2Ø 2Ø 12 ad cØ cØ cØ  79
11a1 : cØ bd 92 Ød Ød ØØ 2Ø cf  77
11a9 : 52 44 45 52 2Ø c4 52 49  bd
11b1 : 56 45 2Ø 23 ØØ 93 11 2Ø  37
11b9 : c3 41 54 41 4c 4f 47 ØØ  b6
11c1 : 2Ø c8 41 52 44 57 41 52  89
11c9 : 45 46 45 48 4c 45 52 2Ø  Ø4
11d1 : 21 21 21 Ød Ød 2Ø c2 49  dc
11d9 : 54 54 45 2Ø 42 45 49 44  a9
11e1 : 45 2Ø cc 41 55 46 57 45  Ø1
11e9 : 52 4b 45 2Ø 41 4e 53 43  91
11f1 : 48 41 4c 54 45 4e 2Ø 21  Ø1
11f9 : ØØ 93 11 2Ø c3 4f 5Ø 59  b6
12Ø1 : 2Ø c6 49 4c 45 53 2Ø 23  16
12Ø9 : 38 2Ø 54 4f 2Ø 23 39 Ød  6a
1211 : Ød 2Ø c2 49 54 54 45 2Ø  45
1219 : d1 55 45 4c 4c 2d 2Ø 28  6f
1221 : 23 38 29 2Ø 55 4e 44 2Ø  c8
1229 : da 49 45 4c 44 49 53 4b  f5
1231 : 2Ø 28 23 39 29 Ød 2Ø 45  5b
1239 : 49 4e 4c 45 47 45 4e 2Ø  7d
1241 : 21 Ød Ød ØØ 93 11 2Ø d3  16
1249 : 43 52 41 54 43 48 2Ø c6  15
1251 : 49 4c 45 53 2Ø 23 38 2Ø  b8
1259 : 3a Ød Ød 2Ø c2 49 54 54  d2
1261 : 45 2Ø d3 43 52 41 54 43  1b
1269 : 48 2d c4 49 53 4b 45 54  ef
1271 : 54 45 2Ø 45 49 4e 4c 45  db
1279 : 47 45 4e 2Ø 21 Ød Ød ØØ  a9
1281 : Ød 91 1d 1d 1d 1d 1d 1d  ab
1289 : 1d 1d 1d 1d 1d 1d 1d 1d  89
1291 : 1d 1d 1d 1d 1d 1d 1d 1d  91
1299 : 1d 1d 1d 1d 1d 1d 1d 1d  99
12a1 : 28 4a 2f 4e 29 3f 9d 9d  c2
12a9 : 9d 9d 9d 9d ØØ 2Ø 12 2Ø  b9
12b1 : ca 41 2Ø 92 2Ø ØØ 2Ø ce  96
12b9 : 45 49 4e 2Ø ØØ Ød 2Ø d3  cb
12c1 : 43 52 41 54 43 48 49 4e  4Ø
12c9 : 47 2Ø 22 ØØ Ød 2Ø c3 4f  28
12d1 : 5Ø 59 49 4e 47 2Ø 22 ØØ  e8
12d9 : Ød 2Ø cf.4b 2Ø 28 4a 2f  1e
12e1 : 4e 29 2Ø 3f Ød Ød ØØ Ød  Ø7
12e9 : 2Ø d3 43 52 45 45 4e 2d  2Ø
12f1 : c2 4c 41 4e 4b 49 4e 47  ba
12f9 : 2Ø 28 4a 2f 4e 29 2Ø 3f  d3
13Ø1 : Ød ØØ Ød 2Ø c3 4f 5Ø 59  ØØ
13Ø9 : 2Ø 43 4f 4d 5Ø 4c 45 54  6d
1311 : 45 2Ø 21 Ød ØØ 85 89 86  bØ
1319 : 8a 87 8b 88 8c Ø3 98 Ø8  ae
1321 : ac Øa 7a Ø8 7f Ø8 84 Ø8  cc
1329 : 89 Ø8 8e Ø8 93 Ø8 75 Ø8  ba
1331 : Ø3 ØØ ØØ 2Ø 2Ø 2Ø 2Ø 2Ø  fc
1339 : 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø  39
1341 : 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø ØØ  Ø1
```

Listing 1. »Dual-Filecopy«. Bitte mit dem MSE eingeben. (Schluß)

Der Formatier-Expreß

Das Vorbereiten von Disketten im 1541-Format ist normalerweise eine extrem zeitraubende Angelegenheit. Dank »TURBO FORMAT« werden diese Zwangspausen nun auf ein Minimum reduziert.

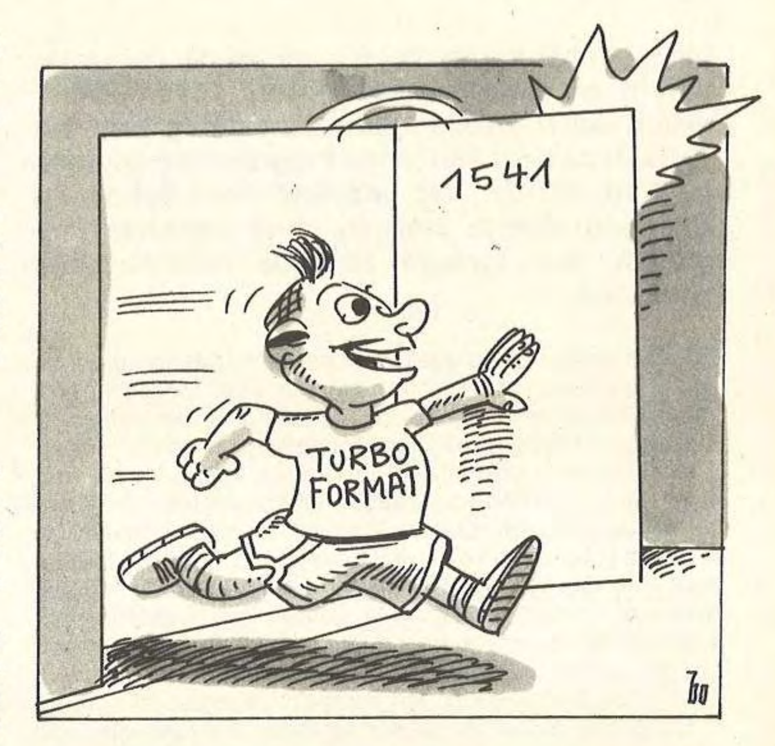
Geschwindigkeit ist nicht gerade eine der Stärken der Floppy 1541. Schuld daran ist das Betriebssystem des C 64 und das DOS der 1541-Diskettenstation, das eine Reihe äußerst zeitintensiver Unterprogramme enthält. Hierzu ist auch die Formatierungs-Routine zu rechnen. Wieviel Zeit hier vom DOS sinnlos verschwendet wird, zeigt ein direkter Vergleich mit »TURBO FORMAT« (Listing 1):

1541-DOS-Format: 77 Sekunden
Turbo Format: 17 Sekunden

Die neue Routine ist also ziemlich genau eine Minute schneller als das Original - und das bei gleicher Leistung. Verzichtet man auf die nachträgliche Überprüfung der Blöcke (VERIFY), benötigt die Floppy sogar nur noch 10 Sekunden für die vollständige Formatierung einer Diskette (Track 1 bis 35).

Turbo Format zeichnet sich neben seiner Geschwindigkeit auch wegen seines Bedienungskomforts aus. So sind keine »OPEN«-Befehle, sondern lediglich der Name und die (neue) ID der Diskette einzugeben. Drücken Sie an dieser Stelle nur <RETURN>, dann werden Name und ID von der vorhergehenden Diskette übernommen.

Die anschließende Frage nach »VERIFY« können Sie mit »J« für Ja und »N« für Nein beantworten. Nach einem Tastendruck beginnt der Formatiervorgang, der mit Ausgabe der Statusmeldung der Floppy endet. Nach erneutem Druck einer beliebigen Taste springt das Programm zurück an den Anfang, und Sie können die nächste Diskette formatieren. Sollte das Programm die Formatierung einmal wegen hardwarebedingter Fehler abbrechen, ist es ratsam, die Floppy aus- und wieder einzuschalten. (Klaus Wenger/nj)




```
Name : turbo-format v01  0801 0b6d
-----------------------------------
0801 : 0b 08 c2 07 9e 32 30 36  4a
0809 : 31 00 00 00 a0 08 a9 c3  b3
0811 : 20 1e ab a9 fb 8d 21 d0  b3
0819 : a2 00 20 cf ff c9 0d f0  22
0821 : 06 9d eb 0a e8 d0 f3 4c  af
0829 : 00 0b 20 83 08 a9 45 20  4a
0831 : dd ed 8a 20 dd ed 98 20  9b
0839 : dd ed 4c ae ff 85 20 84  ab
0841 : 21 20 83 08 a9 57 20 dd  e6
0849 : ed a5 20 20 dd ed a5 21  3b
0851 : 20 dd ed a9 20 20 dd ed  67
0859 : a0 00 b1 30 20 dd ed c8  a6
0861 : c0 20 90 f6 20 ae ff a5  f7
0869 : 20 18 69 20 85 20 90 02  93
0871 : e6 21 a5 30 18 69 20 85  b0
0879 : 30 90 02 e6 31 c6 32 d0  02
0881 : c0 60 a9 08 20 0c ed a9  4a
0889 : 6f 20 b9 ed a9 4d 20 dd  76
0891 : ed a9 2d 4c dd ed a9 09  2e
0899 : 85 31 a9 00 85 30 a2 10  a6
08a1 : 86 32 a0 04 20 3e 08 a2  42
08a9 : 00 a0 04 20 2b 08 20 23  b8
08b1 : 0b a9 00 85 90 20 13 ee  76
08b9 : 20 16 e7 a5 90 f0 f6 4c  98
08c1 : 3b 0b 90 93 0d 20 54 55  e6
08c9 : 52 42 4f 2d 46 4f 52 4d  79
08d1 : 41 54 20 56 4f 4e 20 4b  8e
08d9 : 4c 41 55 53 20 57 45 4e  f4
08e1 : 47 45 52 20 31 39 38 36  8e
08e9 : 0d 0d 20 4e 41 4d 45 2c  3b
08f1 : 20 49 44 20 45 49 4e 47  31
08f9 : 45 42 45 4e 3a 20 00 78  10
0901 : a9 de 8d 00 1c a2 14 8e  c1
0909 : 05 1c ca 8e 74 02 bd ea  c5
0911 : 05 9d ff 01 ca d0 f7 e8  ea
0919 : a0 4a 84 3a c8 8c 00 07  46
0921 : 8a 9d 00 07 e8 d0 fa a9  af
0929 : 07 85 31 20 8f f7 a2 bb  fe
0931 : bd 00 01 9d 00 06 e8 d0  58
0939 : f7 20 a6 c6 ac 4b 02 b9  63
0941 : 01 02 95 12 b9 02 02 85  a6
0949 : 13 a5 22 d0 02 a9 23 0a  e0
0951 : a8 20 34 fa 20 b4 05 88  43
0959 : d0 f7 c8 84 22 ad 00 1c  af
0961 : 29 fc 8d 00 1c a5 22 20  23
0969 : 4b f2 85 43 10 02 a2 02  97
0971 : ad 00 1c 29 9f 1d e7 05  d7
0979 : 8d 00 1c a2 ff 8e 03 1c  1a
0981 : a9 ce 8d 0c 1c a9 55 8d  f6
0989 : 01 1c e8 86 32 86 08 a5  66
0991 : 39 9d 00 03 a5 08 9d 02  0e
0999 : 03 a5 22 9d 03 03 a5 13  b0
09a1 : 9d 04 03 a5 12 9d 05 03  de
09a9 : a9 0f 9d 06 03 a5 08 45  0a
09b1 : 22 45 12 45 13 9d 01 03  cb
09b9 : e6 08 8a 20 d5 d2 a5 43  5b
09c1 : c5 08 d0 cb 8a 48 a9 03  cf
09c9 : 85 31 20 30 fe 68 a8 88  dc
09d1 : 20 e5 fd 20 f5 fd 20 d6  e5
09d9 : 05 a2 0a a4 32 50 fe b9  5c
09e1 : 00 03 b8 8d 01 1c c8 ca  ec
09e9 : d0 f3 84 32 a2 09 20 db  c5
09f1 : 05 20 d6 05 a2 bb 50 fe  a4
09f9 : b8 bd 00 06 8d 01 1c e8  74
0a01 : d0 f4 a0 00 50 fe b8 b9  c7
0a09 : 00 07 8d 01 1c c8 d0 f4  46
0a11 : a2 09 20 db 05 c6 08 d0  04
0a19 : bd 50 fe b8 50 fe b8 20  f5
0a21 : 00 fe a9 00 d0 02 f0 45  76
0a29 : 85 0a 86 30 a5 43 85 08  f6
0a31 : 20 be 05 a2 0a a4 30 50  6d
0a39 : fe b8 ad 01 1c d9 00 03  b5
0a41 : d0 42 c8 ca d0 f1 84 30  cd
0a49 : 20 be 05 a0 bb 50 fe b8  c9
0a51 : b9 00 06 4d 01 1c d0 2c  c2
0a59 : c8 d0 f2 50 fe b8 ad 01  be
0a61 : 1c d9 00 07 d0 1e c8 d0  0e
0a69 : f2 c6 08 d0 c3 a5 22 c9  60
0a71 : 23 f0 1e e6 22 20 65 fa  1f
0a79 : 20 b4 05 20 65 fa 20 b4  51
0a81 : 05 4c 66 04 c6 0a d0 a4  10
0a89 : a5 22 85 80 a9 08 4c 45  47
0a91 : e6 a0 22 20 34 fa 20 b4  59
0a99 : 05 88 d0 f7 a9 12 85 22  9b
0aa1 : c8 8c 07 1c 58 20 40 ee  5a
0aa9 : a2 3a 20 b6 05 a9 3a 8d  e9
0ab1 : 07 1c 60 a2 03 e6 30 d0  fd
0ab9 : fc ca d0 f9 60 a9 d0 8d  3f
0ac1 : 05 18 2c 05 18 10 0a 2c  01
0ac9 : 00 1c 30 f6 ad 01 1c b8  87
0ad1 : 60 68 68 d0 af a2 05 a9  11
0ad9 : ff 0c a9 55 50 fe b8 8d  ee
0ae1 : 01 1c ca d0 f7 60 00 20  80
0ae9 : 40 60 54 55 52 42 4f 2d  e8
0af1 : 46 4f 52 4d 41 54 20 56  01
0af9 : 30 31 2c 38 36 00 00 a9  8b
0b01 : 2d a0 0b 20 1e ab 20 e4  cf
0b09 : ff f0 fb c9 4a d0 08 a9  57
0b11 : 64 8d 24 0a 4c 46 0b c9  3d
0b19 : 4e d0 eb a9 00 8d 24 0a  10
0b21 : f0 f2 a9 08 20 09 ed a9  4b
0b29 : 6f 4c c7 ed 0d 0d 20 56  d4
0b31 : 45 52 49 46 59 3f 0d 0d  98
0b39 : 20 00 20 ae ff 20 e4 ff  cc
0b41 : f0 fb 4c 0d 08 a9 55 a0  48
0b49 : 0b 20 1e ab 20 e4 ff f0  6c
0b51 : fb 4c 97 08 44 49 53 4b  cc
0b59 : 20 45 49 4e 4c 45 47 45  cf
0b61 : 4e 2c 20 54 41 53 54 45  e3
0b69 : 0d 0d 20 00 a9 07 8d 01  10
```

Listing 1. Geben Sie das Programm »TURBO FORMAT« bitte mit dem MSE ein

Die Datasette str

Einer der häufigsten Fehler, der bei der Datasette auftritt, ist ein verstellter Tonkopf. Dieser Fehler macht sich besonders dann bemerkbar, wenn mit Turbo Tape oder ähnlichen Programmen gearbeitet wird. Mit der hier beschriebenen Schaltung läßt sich extrem einfach, ohne jegliches Programm, der Tonkopf an jede Datenkassette anpassen.

Um die Datasette oder einen anderen Datenrecorder zu justieren, gibt es grundsätzlich zwei Möglichkeiten. Eine kleine elektronische Schaltung, mit der sich unabhängig vom Computer die Tonkopfstellung an jede Datenkassette anpassen läßt und ein Programm, das in irgendeiner Form die Tonkopfstellung grafisch auf dem Monitor des Computers darstellt. Ein solches Programm ist aber unbrauchbar, egal wie gut oder schlecht es ist. Der Grund dafür ist ganz einfach der, daß sich nach erfolgter Justage Programme, die zuvor auf anderen Kassetten gespeichert wurden, nicht mehr laden lassen; unter anderem auch das Justageprogramm selbst. Sollen solche Programme geladen werden, müßte das Justageprogramm noch einmal abgetippt werden.

Um das zu vermeiden, stellen wir Ihnen eine Schaltung vor, mit der das Einstellen extrem einfach wird.

Damit die Schaltung verständlich wird, zuerst ein paar Worte zur Datasetten-Elektronik.

Theorie und Praxis

Sie besteht aus zwei Hauptgruppen, einem zweistufigen Verstärker, der die Aufgabe hat, das analoge Signal, das vom Tonkopf kommt, zu verstärken.

Analog deshalb, weil sich digitale Signale nicht auf Band speichern lassen. Selbst wenn ein solches Signal am Tonkopf anliegt, wird es nicht als solches auf das Band geschrieben, sondern in Form einer Sinusschwingung. Beim Laden muß diese Sinusschwingung wieder in eine Form gebracht werden, die der Computer versteht. Folglich muß die Sinusschwingung in ein Rechtecksignal gewandelt werden.

Dies geschieht in der zweiten Hauptstufe mit Hilfe eines Schmitt-Triggers. Am Ausgang des Schmitt-Triggers liegt das Signal in Form einer Rechteckschwingung vor, die entweder einen Spannungspegel von 0 oder 5 Volt hat. Dieses Signal eignet sich nicht zur Einstellung des Tonkopfes, weil die Amplitude des Signals, unabhängig von der Tonkopfstellung, immer konstant zwischen 0 und 5 Volt hin- und herspringt.

Die Messung mit einem Oszilloskop ergab aber, daß, abhängig von der Tonkopfstellung, die Amplitude der analogen Spannung schwankte.

Ist der Tonkopf optimal eingestellt, geht die Amplitude der Spannung gegen ein Maximum. Ist der Tonkopf dejustiert, weicht die Amplitude, abhängig von der Tonkopfstellung, vom Maximum ab. Man kann es jedoch keinem Datasetten-Besitzer zumuten, sich ein Oszilloskop anzuschaffen, nur um die Datasette zu justieren.

Die vorliegende Bastelanleitung, deren Bauteile zu einem Preis von unter fünf Mark zu haben sind, ersetzt in diesem Fall ein Oszilloskop. Mit der Schaltung (Bild 1) läßt sich eine Spannung, natürlich in gewissen Grenzen, auf Maximum abgleichen. Das Herz ist ein Operationsverstärker vom Typ LF 356, der als Komperator (Schwellwertschalter) betrieben wird. Außerdem hat dieser Operationsverstärker gegenüber anderen den Vorteil, daß seine Eingangsstufe aus einem Feldeffekttransistor besteht. Der Eingangswiderstand geht dadurch gegen unendlich und belastet das zu messende Signal in keinster Weise. Mit dem Trimmpotentiometer läßt sich eine Schwellspannung (Bild 2) einstellen, die laufend mit der analogen Sinusschwingung verglichen wird.

Ist der Momentanwert der Sinusschwingung kleiner als die vorgegebene Schwellspannung, führt der Ausgang des LF 356 0 Volt. Wird der Momentanwert größer, springt der Ausgang des LF 356 auf +5 Volt und regt dadurch eine Leuchtdiode an. Wird die Schwellspannung in den Scheitelpunkt der Sinusschwingung gelegt (gestrichelte Linie in Bild 2), geht die Zeitspanne, in der der Ausgang des Komperators auf 5 Volt liegt, gegen ein Minimum. Daraus folgt, daß die Helligkeit der Leuchtdiode abnimmt, je näher die Schwellspannung an den Scheitelwert der Sinusschwingung rückt. Wird dagegen die Amplitude des Signals, also der Sinusschwingung,

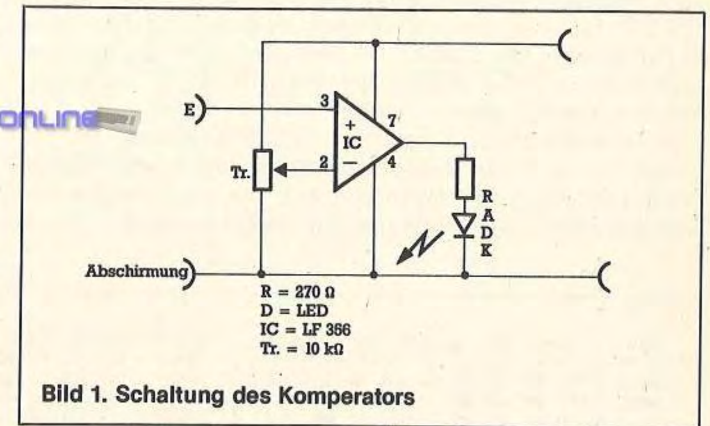


Bild 1. Schaltung des Komperators

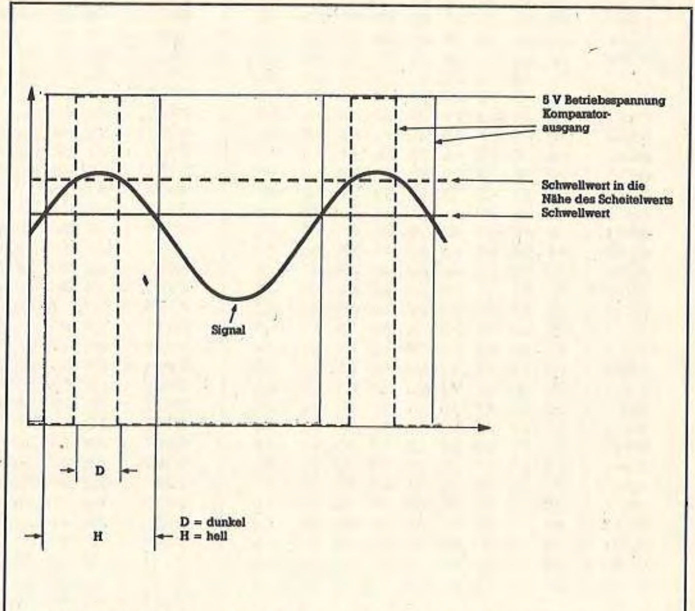


Bild 2. Die Helligkeit der Leuchtdiode ist abhängig von der Zeitspanne, die der Komperatorausgang auf +5V liegt.

eikt nie wieder

vergrößert, wird die Helligkeit der Leuchtdiode wieder größer. Denn die Zeitspanne, in der der Ausgang des Komperators auf 5 Volt liegt, vergrößert sich. Dieses ist vom Prinzip her der ganze Abgleichvorgang. Mit dem Trimmpotentiometer wird auf minimale Helligkeit und mit der Tonkopfeinstellschraube auf maximale Helligkeit abgeglichen.

Aufgebaut wird die Schaltung auf einer kleinen Lochrasterplatine. Diejenigen, die sich eine Platine ätzen wollen, finden das Layout im Verhältnis 1:1 in Bild 3. Wie die einzelnen Pins der Bauelemente miteinander verbunden werden, zeigt Bild 4. Achten Sie beim Zusammenbau auf die richtige Polarität der Leuchtdiode (Bild 5).

Ist die Schaltung zusammengelötet, muß sie noch im Datasettengehäuse untergebracht werden. Öffnen Sie dazu die Datasette und bohren an einer geeigneten Stelle ein Loch in das Gehäuseoberteil, so daß die Leuchtdiode gerade in dieses Loch paßt.

Verbinden Sie die Anschlüsse »+« und »–« (Bild 4) mit den Motoranschlußklemmen. Dabei ist ebenfalls auf die Polarität zu achten. Im allgemeinen ist sie auf dem Motor gekennzeichnet. Der in Bild 4 gekennzeichnete Punkt »E« (für Eingang) muß über ein abgeschirmtes Kabel mit einem der beiden Lötpunkte auf der Datasettenplatine (Bild 6) verbunden werden. Die Abschirmung ist an den mit »–« gekennzeichneten Punkt (Bild 4) zu löten.

Bei den beiden Lötpunkten handelt es sich um den Ausgang des ersten beziehungsweise zweiten Analogverstärkers einer Commodore-Datasette.

Geräte anderer Hersteller sind zum Teil anders aufgebaut. Es kann vorkommen, daß die beiden in Bild 6 gekennzeichneten Analogverstärker in einem Gehäuse untergebracht sind. In diesem Fall ist der Punkt »E« mit dem Pin 8 dieses ICs zu verbinden.

Bevor die Datasette zusammengebaut wird, ist die Schaltung an die Datasetten-Elektronik anzupassen. Schalten Sie dazu den C 64 ein, legen eine Programm-Kassette in die Datasette und drücken die PLAY-Taste.

Nun muß in einem wechselseitigen Einstellvorgang die Helligkeit der Leuchtdiode am Trimmpotentiometer auf Minimum und an der Tonkopfeinstellschraube auf Maximum abgeglichen werden.

Bei Commodore-Datasetten befindet sich die Tonkopfeinstellschraube (Kreuzschlitz) bei gedrückter PLAY-Taste unter einem etwa 5 mm großen Loch auf dem Gehäuseoberteil.

Soll eine andere Datasette justiert werden, muß der Kassettendeckel abgebaut werden. Die Tonkopfschraube ist nun eine der beiden Tonkopfbefestigungsschrauben und zwar die, an der sich eine Spiralfeder befindet. Doch nun zum Abgleichvorgang. Dazu gehen Sie bitte folgendermaßen vor:

1. Am Trimmpotentiometer drehen, bis die Leuchtdiode schwach flackert. Dadurch wird die Schwell- oder Schaltspannung in den Scheitelpunkt der Sinusschwingung gelegt.

2. An der Tonkopfeinstellschraube drehen, bis die Helligkeit der Leuchtdiode ein Maximum erreicht hat. Dadurch wird die Amplitude des Signals, das vom Tonkopf kommt, auf Maximum abgeglichen.

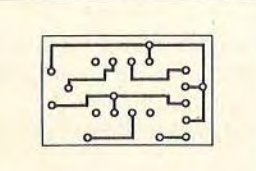


Bild 3. Layout im Maßstab 1:1 (Lötseite)

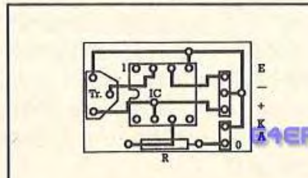


Bild 4. Bestückungsplan (Lötseite)

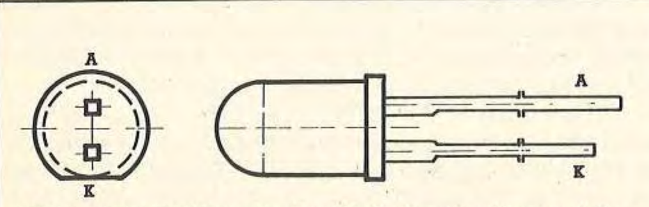


Bild 5. Beim Einlöten der Leuchtdiode unbedingt auf die Polarität achten. Anode=A=längeres Beinchen oder runde Seite.

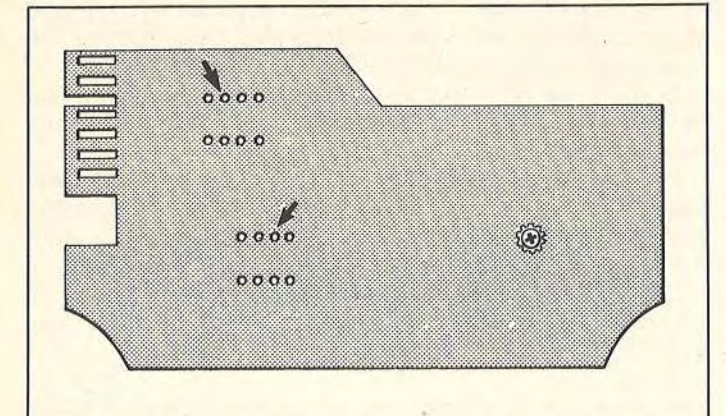


Bild 6. An einen der gekennzeichneten Punkte ist der Punkt »E« (Bild 4) zu löten.

Noch ein Tip

Der letzte Punkt ist nur dann erforderlich, wenn die Datasette nicht optimal eingestellt war, beziehungsweise eine Kassette benutzt wird, die mit einer anderen Datasette beschrieben wurde.

In diesem Fall muß der Einstellvorgang so lange wiederholt werden, bis eine Einstellung erreicht ist, bei der die Leuchtdiode erlischt, sobald der Tonkopf minimal verstellt wird. Bauen Sie nun die Datasette wieder zusammen. Schalten Sie vorher aber den C 64 aus.

Wollen Sie jetzt ein Programm laden, das mit einem dejustierten Tonkopf aufgenommen wurde, brauchen Sie nur noch, nachdem der C 64 eingeschaltet wurde, die Kassette einzulegen, die PLAY-Taste zu drücken und so lange an der Tonkopfeinstellschraube zu drehen, bis die Helligkeit der Leuchtdiode ein Maximum erreicht hat.

Zum Schluß soll noch darauf hingewiesen werden, daß selbst bei Commodore-Datasetten die unterschiedlichsten Platinen existieren. Befinden sich auf Ihrer Platine nur zwei 14beinige ICs, dann ist der Punkt »E« an den Pin 8 oder 13 jenes ICs zu löten, das sich auf der linken Platinenseite befindet (vorausgesetzt, Sie haben die Platine so vor sich liegen, wie Bild 6 zeigt). (ah)

64'er-DOS erweitert

Auch Gutes läßt sich noch verbessern. Das neue 64'er-DOS Version 4 (EX-SMON-DOS) enthält zusätzlich einen komfortablen Monitor, Basic-Toolkit-Befehle und eine Bildschirm-Hardcopy.

Schon wieder ein neues Kernel? werden viele jetzt fragen, und tatsächlich: Wer geglaubt hatte, das »64'er-DOS V3« sei wohl der letzte Schritt des (sinnvollen) Betriebssystemausbaus für den C 64, wird überrascht sein: Das neue System beinhaltet nicht nur das komplette 64'er DOS V3, sondern eine zusätzliche Fülle von Programmierhilfen, wie sie zum Teil auch in größeren Computern vergeblich gesucht werden.

Sie werden in Kürze feststellen, daß Sie für das bißchen Geld und wenig Aufwand einen neuen, wirklich besseren Computer bekommen haben. Erst beim späteren Umschalten auf das Originalsystem merkt man richtig, daß dieses viel eher dazu angetan ist, einem das Computern zu verleiden, anstatt Freude zu bereiten.

Noch eine kurze Meldung: Nach Redaktionsschluß erreichte uns noch eine Erweiterung zum DOS V4: Es enthält nun auch als Ersatz der RS232-Schnittstelle eine komfortable Centronics-Schnittstelle, die am User-Port nach außen führt. Es war uns leider aus Zeitgründen nicht mehr möglich, das neue Listing und die erreichte Anleitung abzudrucken. Die als »CENT-SMON-Kernel« bezeichnete Datei sowie die Anleitung zur Centronics-Schnittstelle sind jedoch auf der zu dieser Ausgabe erscheinenden Programmservice-Diskette enthalten. Bis auf die geänderte Software, die zum Brennen des Kernels benötigt wird, ändert sich jedoch nichts an der nun folgenden Beschreibung des neuen Betriebssystems.

Das vorliegende, erweiterte »EX-SMON-DOS« ist für diejenigen gedacht, die mit ihrem Computer nicht nur spielen, sondern auch selbst programmieren. Alle hier nicht erwähnten Funktionen des 64'er-Kernel V3 (außer »SYS 4096*«) sind vollständig vorhanden und funktionieren gemäß dessen Beschreibung in Ausgabe 3/86.

EX-SMON-DOS ist zum 64'er-DOS V3 vollständig kompatibel und benötigt – wie jenes – den Einbau je eines neuen EPROMs im Computer und in der Floppy. Dazu ist bei diesem System eine Lötstelle erforderlich, die aber auch der Laie mit etwas Sorgfalt problem- und vor allem gefahrlos machen kann. Eventuell vorhandene Sockel oder Umschaltplatinen können weiter verwendet werden.

Weiter unten wird der Einbau noch genauer beschrieben, vorher aber wollen wir doch sehen, was das EX-SMON-DOS alles kann:

Die neue Belegung der Funktionstasten

Die Funktionstasten wurden gegenüber dem 64'er-Kernel V3 etwas anders belegt, um die Bedienung so angenehm wie möglich zu machen. Sie erfüllen nun folgende Aufgaben:

<F1> gibt das Directory auf dem Bildschirm ohne Programmverlust aus

<F2> LOAD

<F3> Bildschirm löschen und RUN.

<F4> MERGE – mit dieser Befehlstaste ist es möglich, ein zweites Basic-Programm an ein schon im Speicher vorhandenes anzuhängen. Dabei kann das zweite Programm auch niedrigere Zeilennummern haben – nur bei gleichen Sprungzielen gibt's bei späterem RENUMBER Probleme, da RENUMBER das jeweils erste Ziel als richtig erkennt.

Benutzt wird MERGE wie folgt: mit <F1> Directory auf den Bildschirm bringen, und dann mit dem Cursor auf die Zeile mit dem gewünschten Programmnamen fahren. Jetzt <F4> drücken, und schon wird das neue Programm einfach hinten an das vorhandene angehängt.

Achtung: Die LIST-Tasten <F5> und <F7> zeigen das zweite Programm nur, wenn dessen erste Zeilennummer höher ist als die letzte des alten! Es kann sonst nur mit dem normalen LIST-Befehl betrachtet werden.

<F5> PAGE-UP – diese Taste listet ab der obersten auf dem Bildschirm stehenden Zeilennummer eine Seite nach »oben« (rückwärts im Basic-Programm!). Wenn keine Nummer auf dem Schirm steht, so wird die erste »Programmseite« gezeigt.

<F6> schreibt »SAVE"« auf den Bildschirm.

<F7> PAGE-DOWN – listet ab der Cursor-Position beziehungsweise ab der untersten Zahl auf dem Bildschirm eine Seite nach »unten«. Statt also LIST 450 <RETURN> einzugeben, kann einfach 450 an den linken Bildrand geschrieben werden, dann wird <F7> betätigt: der Computer listet ab Zeile 450 bis Bildschirmende.

<F5> und <F7> führen zudem automatisch OLD durch (Wiederherstellen eines durch NEW oder RESET gelöschten Programms), wenn kein Programm gefunden wird.

<F8> schaltet um zwischen Floppy-Geräteadresse #8 oder #9.

<CTRL+F1> führt in den COMMAND-Modus (siehe unten).

<CTRL+CBM+F5> diese Fingerakrobatik-Tastenkombination hat ihren Grund: Sie führt einen RESET durch, was ja nicht versehentlich geschehen soll. Außerdem ist sie (im Gegensatz zu den anderen Funktionen) auch bei vielen laufenden Programmen in Betrieb. Diese Tastenkombination wird von keinem uns bekannten Programm benutzt.

<CTRL+CBM+F7> PRINT-SCREEN – auch diese Funktion ist aus dem laufenden Programm abrufbar: Sie gibt jederzeit eine Hardcopy des aktuellen Text-Bildschirms auf den Drucker! Das Programm macht dabei einfach keine Pause und läuft nach erfolgtem Druck unbeirrt weiter!

Der Drucker muß am seriellen Port mit Device #4 vorhanden sein. Um auch Commodore-fremde Drucker (mit Interface) bedienen zu können, wird auf die Ausgabe des Reverszeichens verzichtet. Zwischen Groß-/Kleinschreibung und Normalgrafik wird mit Sekundäradresse 7 beziehungsweise 0 umgeschaltet (dem Bildschirm entsprechend).

Bevor diese Routine anläuft, werden etliche Sicherheitsabfragen gemacht, da nur der Original-Commodore-Zeichensatz ausgegeben wird.

PRINT-SCREEN funktioniert mit praktisch allen Basic- und vielen Maschinenprogrammen. Es kann jedoch Programme geben, die bei Betätigung von <CTRL+CBM+F7> hoffnungslos abstürzen!

Der COMMAND-Modus stellt eine ganze Reihe nützlicher Editor-Routinen zur Verfügung. Nach Druck von <CTRL+F1> fragt der Computer:

```
COMMAND ?
```

und der Cursor blinkt zum Zeichen, daß eine Eingabe erwartet wird. Er reagiert (ohne <RETURN>) auf folgende Tasten:

<A> AUTO-NUMBER – zur leichteren Eingabe eines Basic-Programmes gibt der Computer nach jedem

<RETURN> die nächste Zeilennummer aus. Es können Schrittweiten von 1 bis 255 angegeben werden. Begonnen wird ab der Nummer der nächsten eingegebenen Zeile. Korrekturen an vorhandenen Zeilen können trotz eingeschaltetem AUTO-NUMBER vorgenommen werden (kein »Zeilenkiller« wie andere...). AUTO-NUMBER läßt sich ausschalten, indem man eine Zeilennummer leer (also ohne weiteren Text) mit <RETURN> eingibt (diese Zeile wird, sofern im Programm vorhanden, wie üblich gelöscht).

<D> DELETE - damit können beliebig lange Programmteile gelöscht werden. Delete fragt nach der ersten (»FROM«) und der letzten (»TO«) Zeilennummer des Programmbereiches, der gelöscht werden soll. Achtung: Mit DELETE gelöschte Zeilen können (außer durch Neueingabe) nicht wiederhergestellt werden! Bei Fehleingaben kann durch Eingabe von 0 bei »TO« ausgestiegen werden.

<F> FIND - dieses Kommando findet eine beliebige Zeichenfolge in einem Basic-Programm und gibt die entsprechende Zeilennummer aus (oder mehrere, falls sich die Zeichenfolge öfters im Programm findet). Bei der Eingabe der zu suchenden Zeichen muß folgendes beachtet werden: Text, der hinter REM oder in Anführungszeichen steht, wird vom C 64 anders behandelt als das übrige Programm. Bei der Suche nach solchen Zeichenfolgen muß als erstes Zeichen ein Anführungszeichen stehen.

<M> MOVE - eine Funktion, die man aus keiner Basic-Erweiterung, sondern nur aus Textverarbeitungsprogrammen kennt: Mit MOVE können ganze Programmblöcke an eine andere Stelle verschoben werden! Wie oft kommt es vor, daß man während des Programmierens eine Routine zum Beispiel ab Zeile 2000 bis 2400 eingegeben hat, sie später aber lieber zwischen Zeile 400 und 410 hätte, weil dadurch der Programmfluß besser ersichtlich wäre! Kein Problem mit MOVE:

Zuerst wird die erste Zeile des zu verschiebenden Blocks (»FROM«, also hier 2000) angegeben.

Die zweite Eingabe bezeichnet das Ende des zu verschiebenden Programmblocks (»TO«): im Beispiel also 2400.

Nun sagen wir dem Computer noch, wohin der Block verschoben werden soll (»WHERE«). Dabei geben wir die letzte Zeilennummer an, die ihren alten Platz behalten soll, also im Beispiel Zeile 400.

Der Block wird nun verschoben, und das Programm fährt gleich weiter mit RENUMBER (siehe unten), weil ja jetzt nach Zeile 400 die Zeilen 2000 bis 2400 und erst darauf Zeile 410 folgen.

Damit das Basic-Programm lauffähig bleibt, wird (wie auch bei RENUMBER) vor der Ausführung von MOVE geprüft, ob alle Sprungziele (für GOTO und GOSUB) vorhanden sind, ansonsten werden die fehlerhaften Zeilen angezeigt und die Operation abgebrochen.

MOVE benutzt als Zwischenspeicher das (von Basic nicht genutzte) RAM unter dem Kernel-ROM.

<O> OFF - schaltet sämtliche Befehlserweiterungen und Funktionen des neuen Systems ab. Die Funktionen RESET und PRINT-SCREEN bleiben erhalten.

<R> RENUMBER - numeriert ein Basic-Programm neu mit gleichmäßiger Schrittweite zwischen 1 und 255. Auch die Startnummer kann eingegeben werden.

Alle Sprungziele im Programm, auch von Befehlen wie GOSUB, RUN oder ON GOTO, werden angepaßt, so daß das Programm lauffähig bleibt. RENUMBER funktioniert auch dann, wenn nach einer hohen Zeilennummer eine niedrigere folgt (zum Beispiel nach MERGE oder MOVE). Die Funktion wird nicht ausgeführt, wenn sich Sprünge zu nicht existierenden Zeilennummern im Programm befinden.

RENUMBER kann, je nach Programmlänge, einige Minuten dauern. Dabei wird jeweils die gerade bearbeitete (alte) Zeilennummer angezeigt.

<V> VARIABLEN-DUMP - zeigt alle aktuellen Variablen und deren Wert nach einem Programmablauf an (keine Arrays!).

Um zu verhindern, daß die Angaben am oberen Bildrand verschwinden, kann <V> auch geSHIFTet eingegeben werden (das EX-SMON-Kernel stoppt das LISTen am Bildschirmende bei gedrückter SHIFT-Taste).

Achtung: Wie nach allen COMMAND-Befehlen kann das Programm danach nicht mit CONT fortgesetzt werden!

<X> EXIT - verlassen des COMMAND-Modus ohne weiteres Kommando.

Andere Eingaben als die oben aufgeführten Tasten bei »COMMAND ?« werden mit einem UNDEF'D STATEMENT ERROR quittiert.

Soviel zu den erweiterten Editor-Funktionen - wer damit erst einmal gearbeitet hat, der gibt EX-SMON-DOS nie wieder her!

Noch eine Kleinigkeit: Nach dem Laden von Maschinenprogrammen hinter das Basic-Ende mußte bisher immer NEW eingegeben werden - lästig, wenn noch ein Basic-Programm im Speicher war. Auch diese Zeiten sind vorbei, in diesen Fällen bleiben die Basic-Zeiger so, wie sie vor dem Laden waren!

Kommen wir jetzt zum zweiten Teil der EX-SMON-DOS-Erweiterungen, der mehr die Maschinensprache-Freaks interessieren wird:

Der Monitor

Der Monitor, der hier eingebaut wurde, entspricht in den wichtigsten Zügen dem aus unseren Ausgaben bekannten SMON. Vollständig geändert wurde in der Hauptsache die Belegung der unteren drei Speicherseiten - der neue SMON benutzt nun ausschließlich den Prozessorstack und die Zeropageadressen $39 bis $48 (abgesehen natürlich von den Kernel-Ein- und -Ausgabefunktionen). Diese Speicherstellen werden - im Gegensatz zu den Kassettenadressen - selten von Maschinenprogrammen benutzt, was gerade der TRACE-Funktion (siehe unten) sehr dienlich ist. Da CONT beim Monitorausgang ohnehin gesperrt ist, spielt die Verstellung der Basic-Zeiger keine Rolle.

Aus Platzgründen entfernt wurden die Routinen »B« (Basic-Data), »K« (Kontrolle) und »=« (Vergleichen). Dafür sind drei wichtige Finessen hinzugekommen:

Der Monitor wird direkt aus dem Basic-Editor angesprochen. Das einzige, was zur Ausführung eines Monitorbefehls nötig ist, ist die Eingabe eines Punktes (».«) als erstes Befehlszeichen. So können zum Beispiel direkt aus dem Basic (wie alle Funktionen nur im Direktmodus) Hex-Zahlen umgerechnet oder Hex-Berechnungen ausgeführt werden. Die Syntax entspricht dabei genau der des SMON (siehe SMON-Anleitungen in den 64'er-Ausgaben 11/84 bis 2/85 oder auch komplett im Assembler-Sonderheft 8/85). So ergibt die Eingabe von ».$0079« <RETURN> die Umrechnung von Hex $79 in den entsprechenden Dezimalwert, ».#49152« <RETURN> hat eine Umrechnung in die Hex-Zahl zur Folge. ».GC000« <RETURN> startet ein Maschinenprogramm ab der Adresse 49152 - dabei entspricht diese Eingabe exakt dem vergleichbaren SYS-Befehl (Register werden NICHT aus der SMON-Registeranzeige, sondern aus $030C bis $030F übernommen!). So ist also endlich auch der Aufruf von Maschinenprogrammen mit Hex-Eingabe möglich!

Ebenfalls neu ist, daß der Kernel-SMON auf ALLE jetzt im C 64 vorhandenen Speicherbereiche zugreifen kann. Die Eingabe von ».R« führt - wie beim SMON - zur Registeranzeige. Dabei wird der routinierte SMON-Benutzer sofort feststellen, daß ein »Register« dazugekommen ist: es ist bezeichnet mit dem Kürzel »CP«. Dieses »Register« zeigt nichts anderes als die momentane Prozessorport-Einstellung, im Nor-

malfall $37. Durch Überschreiben dieses Registers wird nun die gewünschte (nur für den Monitor gültige) Konfiguration eingestellt. Folgende Werte sind für dieses Register sinnvoll:

- $ 37 Die beim Einschalten des C 64 vorliegende, normale Speicherkonfiguration (beide ROMs aktiv)
- $ 36 zeigt normales Kernel ab $E000, jedoch RAM ab $A000 (Basic abgeschaltet)
- $ 35 Kernel und Basic abgeschaltet, die I/O-Register werden jedoch angezeigt
- $ 34 zeigt den vollen RAM-Bereich des C 64, also alle Bereiche unter den ROMs und auch unter den I/O-Registern
- $ 33 zeigt den Zeichensatz von $D000 bis $DFFF
- $ 27 zeigt den neuen (»linken«) Kernel-Bereich zwischen $E000 und $FFFF.

So ist es endlich möglich, sich einmal das riesige Speicherloch unter den ROMs auf dem Bildschirm anzusehen – man bekommt plötzlich eine ganz andere Beziehung dazu!

Die letzte – und wohl bemerkenswerteste – SMON-Erweiterung ist eine eigens für dieses System entwickelte Maschinensprache-TRACE-Routine, welche die schrittweise Verfolgung von fast allen Programmen ermöglicht. Auch dieser SMON-Teil arbeitet in allen ROM- und RAM-Bereichen – wenn ein Programm also während des TRACEns den Prozessorport umschalten will, so soll es das ruhig tun, TRACE wird dadurch nicht behindert! Dies wird erreicht durch die Einrichtung eines vollständig softwaregesteuerten Schein-Prozessors, welcher jeden Befehl vor der Ausführung analysiert und die nötigen Angaben auf den Bildschirm bringt. Der Prozessorsimulator führt alle legalen (und nur die!) Befehle des 6510 richtig aus. Er verwaltet einen eigenen Stack, so daß sich gewöhnlich kein Unterschied zur normalen Programmbearbeitung ergibt. Bei Erreichen eines illegalen Opcodes springt er jedoch zurück in die Registeranzeige des Monitors. Hier können übrigens nicht nur die Register, sondern sogar der Stack-Pointer für TRACE verändert werden!

TRACE wird – wie könnte es anders sein – angesprochen durch ».T«+Adresse. Wird keine Adresse eingegeben, so ist der aktuelle Programmzählerstand maßgebend (Achtung, dieser ist beim jeweils ersten Einstieg in den Monitor zufällig!).

Als erstes fragt TRACE nun: »REALTIME IN ROM? Y/N«. Die Eingabe von »Y« bewirkt, daß Subroutinen im ohnehin bekannten ROM-Bereich automatisch in Echtzeit ausgeführt werden. Dies funktioniert jedoch nur, solange beide Original-ROM-Bereiche (Basic und Kernel) eingeschaltet sind.

Jetzt erscheint auf dem Bildschirm die erste Befehlszeile: revers alle Register und der Programmzähler, dahinter der nächste auszuführende Befehl. Am Zeilenende gibt es wieder ein reverses Feld, welches folgende Angaben enthält:

Normalerweise zeigt es die Adresse des nächsten zu bearbeitenden Befehls, gekennzeichnet mit einer spitzen Klammer (>). Bei indirekten JMP-Befehlen erscheint hier die tatsächlich anzuspringende Adresse, angezeigt durch einen Hochpfeil (»↑«). Für BRANCH-Befehle wird die Bedingung geprüft. Ist sie erfüllt, so wird dies durch »=« gekennzeichnet, ansonsten erfolgt die Ausgabe der spitzen Klammer (>). Der Clou ist jedoch die Anzeige bei indizierten (direkt oder indirekt) Lade- oder Speicherbefehlen: Dieses Feld zeigt die tatsächlich bearbeitete Adresse an! So kann beim TRACEn eines Programms ohne große Rechnerei verhindert werden, daß zum Beispiel Interrupt-Register, Vektoren oder andere Speicherstellen beschrieben werden, die eine Weiterführung des Einzelschrittmodus gefährden würden. Wo normalerweise mit <SPACE> der aktuelle Befehl ausgeführt und zum nächsten gegangen wird, kann hier einfach mit »N« der fragliche Befehl übersprungen werden (ist bei allen Befehlen möglich) – der nächste Befehl wird ohne Ausführung des aktuellen erreicht.

Zwei weitere Erleichterungen für die Fehlersuche werden von TRACE zur Verfügung gestellt:

Bekannt ist der JUMP-Befehl (»J«), welcher die aktuelle Subroutine bis zum gültigen RTS in Echtzeit abarbeitet. »J« wird dabei nicht ausgeführt, wenn durch eine Veränderung des Stack seit dem letzten JSR der Rücksprung gefährdet wäre.

Neu – und wohl nur dank des simulierten Prozessors möglich – ist ein QUICK-TRACE-Modus, der sich gewaschen hat: Nach der Eingabe von »Q« verlangt TRACE eine Abbruchbedingung, die eines der folgenden vier Formate haben kann:

X = FB	QUICK-TRACE stoppt, wenn das X-Register den Wert $FB annimmt
Y = 00	Anhalten, wenn YR = $00
P = C412	hält an, falls der aktuelle Programmzähler auf Adresse $C412 zeigt
M4125 = E0	QUICK-TRACE stoppt, sobald die Speicherstelle $4125 auf den Wert $E0 gesetzt wird.

Nach dem Anhalten kann einfach mit <SPACE> im Einzelschrittmodus weitergemacht werden. QUICK-TRACE gibt alle Angaben auf dem Bildschirm (oder Drucker bei Eingabe von <SHIFT+T>) aus, es berücksichtigt außerdem auch die Realtime-Angabe vom Anfang.

Aussteigen kann man aus TRACE mit der Taste <RUN/STOP> – erfolgt dies aus dem Einzelschrittmodus, so kann durch Eingabe von ».T« <RETURN> am gleichen Ort weitergemacht werden.

Eine Bemerkung noch zur SAVE-Routine des neuen SMON: Da auch von diesem Programmteil auf alle Speicherbereiche zugegriffen wird (abhängig vom CP-Register), mußte die Ausgabe über den OPEN-Befehl gelegt werden. Aufgrund des bekannten Floppyfehlers bei PRG-REPLACE-Funktionen sollte unbedingt auf die Verwendung des Klammeraffen verzichtet werden!

Der generelle Ausstieg aus dem Monitormodus erfolgt NICHT durch »X«, sondern einfach durch ein leeres <RETURN>. Auch bei falschem Eingabeformat (ungültige Adreßangaben...) springt das Programm jeweils zurück in den READY-Modus.

Selbstverständlich zeigt auch der BRK-Vektor des erweiterten C 64 nicht mehr auf die Initialisierungsroutine, sondern auf den Monitor.

Eine Übersicht der neuen Befehle ist aus Tabelle 1 ersichtlich. Nähere Informationen zum SMON können Sie entweder in den Ausgaben 11/84 bis 2/85 oder gesammelt dem Assembler-Sonderheft 8/85 entnehmen.

Der Einbau des EX-SMON-DOS

»EX-SMON-DOS« (Kernel-Teil) ist auf Disk 65 Blocks lang (Name des Files auf der Programmservice-Diskette: EX-SMON-KERNEL) und füllt genau ein 16-KByte-EPROM (27128). Dies bedeutet eine Gesamtspeichererweiterung für den C 64 um 8 KByte. Da der zusätzliche Speicherplatz vom Computer nicht ohne weiteres benutzt werden kann, muß eine der (ohnehin nicht gebrauchten) Leitungen für das Kassettengerät als Schalter herhalten. Dazu gleich eine Anmerkung: Mit dem Einbau des neuen Betriebssystems verlieren Sie die Möglichkeit, mit der Datasette arbeiten zu können. Sämtliche Kassettenroutinen wurden gegen die neuen Schnelladeroutinen ausgetauscht, sind also nicht mehr vorhanden.

Um das neue System zu erhalten, benötigen Sie zuerst ein EPROM vom Typ 27128 sowie einen Adaptersockel, der wie aus Bild 1 ersichtlich verdrahtet sein sollte. (Für die Besitzer der DOS-V3-Versionen aus unserem Hardware-Service – die Adaptersockel sind identisch.)

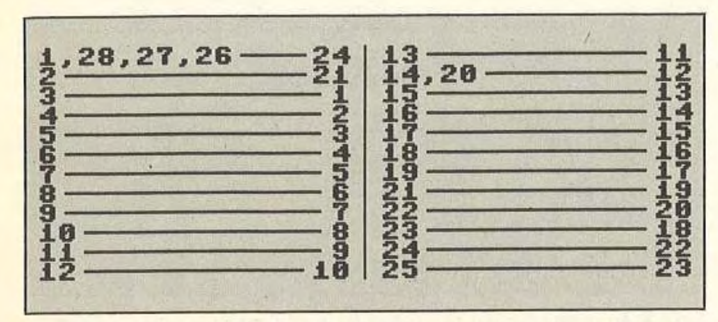


Bild 1. So müssen Sie einen 24- und einen 28poligen Sockel verbinden, um das EPROM in die Steckplätze einzusetzen

Die Software, die gebrannt werden muß, ist in Listing 1 (KER V4...) enthalten. Bitte geben Sie das Programm mit dem MSE ein und speichern es.

Nach dem Laden starten Sie das Programm mit »RUN«. Nach kurzer Zeit, in der die Daten entpackt werden, löst das Programm einen Reset aus und kehrt in den Direktmodus zurück. Die zu brennenden Daten stehen nun im Bereich von $2000 (8196) bis $5FFF (24575). Speichern Sie bitte diesen Bereich mit einem Monitor auf Diskette (denken Sie daran, daß fast jeder Monitor die Angabe ENDE+1 als Endadresse benötigt). Nun können Sie Ihren Eprommer einstecken, die Software laden und das EPROM brennen.

Nehmen Sie das programmierte EPROM in die Hand, Füße nach unten, die Kerbe von sich weggerichtet. Der erste Anschluß links neben der Kerbe ist Pin 1, rechts der Kerbe ist Pin 28 (es wird rundherum gegen den Uhrzeigersinn gezählt). Somit ist Anschluß 26 der dritte Pin rechts, von der Kerbe aus gesehen (siehe auch Bild 2). Dieser Pin 26 muß vorsichtig in die Waagerechte gebogen werden. Machen Sie das langsam und passen Sie auf, daß Sie nicht die danebenliegenden Füße verbiegen.

Sollten Sie noch nicht über das 64'er-DOS V3 (in der Floppy) verfügen, so müssen Sie zuerst noch das neue DOS V3 brennen. Wie das gemacht wird, lesen Sie bitte etwas weiter unten im Text. Ansonsten können Sie nun hier weitermachen.

Öffnen Sie Ihren – vollständig vom Netz getrennten – C 64 (drei Schrauben im Gehäuseboden).

Achtung: Garantieverlust

Setzen Sie, wie schon bei der Einbauanleitung für das 64'er-DOS (Ausgabe 3/86) beschrieben, das EPROM in den Adaptersockel und den Sockel anstelle des Original-Kernel-ROM (Steckplatz U4) in den Computer. Auch hier muß die EPROM-Kerbe von Ihnen weg-, also zum Kassettenport hinzeigen. Der herausgebogene Fuß schaut nun verloren über dem Sockel heraus.

Schalten Sie Ihren Lötkolben ein (am besten ein 16-Watt-Lötkolben) und schneiden Sie sich ein etwa 10 Zentimeter langes Stück dünnen, isolierten Draht (besser Litze) zurecht. Auf beiden Seiten werden nur zirka 2 Millimeter der Isolation abgenommen und beide Drahtenden sauber verzinnt. Ebenfalls sollten Sie schon ein klein wenig Zinn ganz außen auf das vorstehende EPROM-Füßchen auftragen. Heizen Sie aber nicht zu lange (maximal drei Sekunden), damit das EPROM nicht zerstört wird. Passen Sie auch auf, daß kein Zinntropfen zwischen die Füßchen oder gar in den Computer läuft.

Verbinden Sie ein Ende des Drähtchens mit dem herausstehenden EPROM-Fuß.

Das freie Drahtende gehört nun an den Anschluß 6 des Kassettenports, welcher ja gleich hinter dem Kernel liegt. Anschluß 6 ist meistens nur mit »6« gekennzeichnet, es ist der Anschluß ganz links des CN3 (nicht des USER-Ports, wohlgemerkt!). Dort ist ein Lötauge vorhanden, das Sie am besten auch zuerst verzinnen, um danach das Drähtchen daran zu befestigen.

Kontrollieren Sie noch mal, ob der Draht auch hält und ob er wirklich nur den Anschluß 6 des Kassettenports mit dem EPROM-Pin 26 verbindet (Bild 2).

Wenn alles in Ordnung ist, ist die Operation schon beendet und der Computer kann wieder zugeschraubt werden.

Sollten Sie das 64'er-Floppy-DOS V3 (wieder gemäß 64'er, Ausgabe 3/86) schon eingebaut haben, dann steht dem Einschalten nichts mehr im Wege.

Verfügen Sie jedoch noch nicht über das neue DOS V3, ist es jetzt an der Zeit, es zu brennen. Dazu benötigen Sie den oben erwähnten Adaptersockel und ein EPROM Typ 2764.

Geben Sie das Listing 2 (DOS V3) mit dem MSE ein und speichern es. Nach erneutem Laden und Starten löst das Programm nach kurzer Zeit einen Reset aus. Die Daten stehen jetzt im Computerspeicher, und zwar im Bereich von $2000 (8196) bis $3FFF (16383). Speichern Sie diesen Bereich (bedenken Sie die Angabe von ENDE+1 beim Monitor) und schließen Sie Ihren Eprommer an.

Ist das EPROM gebrannt, stecken Sie es in den Adaptersockel und setzen es in der Floppy in den Steckplatz UAB4 ein (die Kerbe zeigt zur Gehäuserückfront). Sollten Sie eine Floppy mit langer Platine besitzen, wäre es der Steckplatz UAB5.

Verfahren Sie, falls nicht schon getan, mit dem EX-SMON-Kernel ebenso (siehe obere Beschreibung). Der Computer und die Floppy können nun geschlossen und in Betrieb genommen werden.

Umschaltmöglichkeit für das Floppy-DOS

Der gleiche Trick mit der Adreßleitung des EPROMs funktioniert auch in der Floppy, nur daß hier mit einem Schalter zwischen Original- und 64'er-DOS umgeschaltet werden kann. Dies ist nur sinnvoll, wenn Sie im Computer eine Umschaltplatine montiert haben (die Sache mit dem Füßchen funktioniert auch dann wie oben beschrieben, der Anschluß stört nicht bei abgeschaltetem System!).

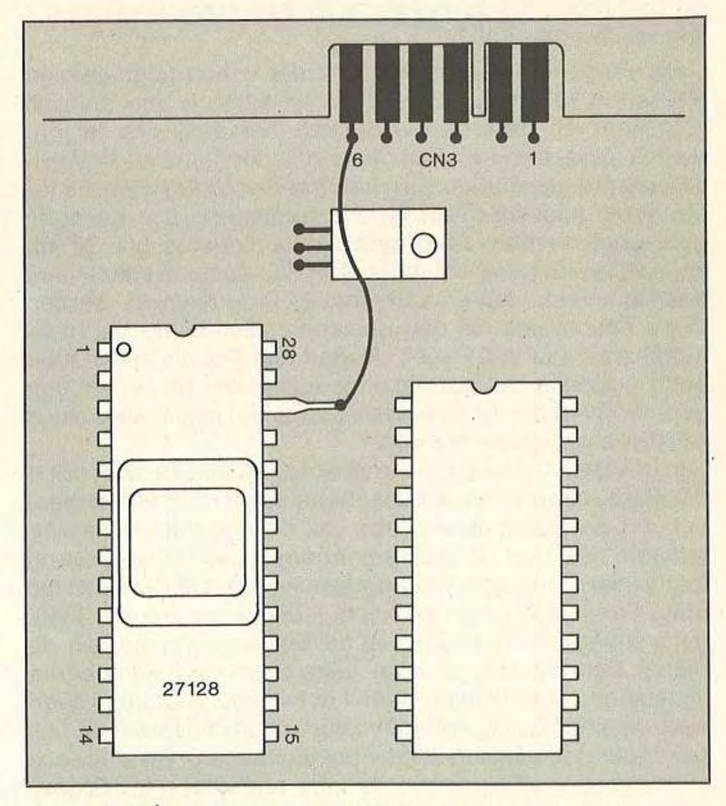


Bild 2. Das freie Kabel (Pin 26) wird am Kassettenport, Anschluß 6, angelötet

Auf der Programmservice-Diskette befindet sich ein zusätzliches File mit dem Namen »DOUBLE DOS«. Es ist ebenfalls 65 Block lang und beinhaltet beide DOS-Versionen, das Original-DOS und das 64'er-DOS V3. Das 64'er-DOS arbeitet dabei neu ohne Disk-Verify, was die Schreibvorgänge um mehr als die Hälfte verkürzt.

Falls Sie sich selbst so ein EPROM mit Umschaltung herstellen möchten, sehen Sie in Bild 3, wie die Daten im EPROM liegen müssen.

Brennen Sie also das »DOUBLE DOS« ebenfalls auf ein 27128-EPROM und biegen Sie, wie gehabt, Pin 26 vorsichtig in die Horizontale. Nachdem das EPROM in den Sockel gesteckt ist (noch nicht in die Floppy!), werden nun an den vorstehenden Pin zwei Dinge angelötet: ein 10 Kilo-Ohm-Widerstand, der (wie aus Bild 4 ersichtlich) am anderen Ende direkt mit Pin 28 (Lötstelle am Sockel) verbunden wird, und ein Stück Kabel (eine von 2 Adern), das später zu einem Schalter führt.

Die zweite Ader des Kabels wird verbunden mit Pin 14, das ist der Ihnen am nächsten liegende Pin links (Kerbe von Ihnen weggerichtet). Das Kabel kann durch einen der Lüftungsschlitze im Boden herausgeführt werden, oder wem es nicht zu schade ist, kann den Schalter auch im Gehäuse einbauen (Achtung, daß das Kabel nicht in den Bereich der Mechanik gerät!).

Ans andere Ende des 2adrigen Kabels kommt ein einfacher Umschalter, an dem die beiden Drahtenden entweder verbunden werden (Original-DOS) oder die Verbindung offen bleibt (64'er-DOS V3).

Jetzt kann einfach zwischen dem 64'er-DOS und dem Original-Commodoresystem umgeschaltet werden (muß natürlich immer gleichzeitig am Computer und der Floppy gemacht werden), womit alle Kompatibilitätsprobleme ein für allemal gelöst sind.

Da wir überzeugt sind, daß es viele Computer- und Elektronik-Freaks gibt, die gern genauer wissen möchten, wie denn das Ganze nun wirklich funktioniert, hier noch etwas schwerere Kost für diese technisch Interessierten:

So umfangreiche Erweiterungen im Kernel eingebaut – wie ist das möglich?

Die Frage ist berechtigt – trotz der schon gestrichenen Kassettenroutinen im 64'er-DOS ist nämlich buchstäblich kein Byte mehr frei in diesem System. Gerade wegen der fehlenden Kassetten-Programmteile ist jedoch etwas viel Wertvolleres frei geworden: Die drei Port-Anschlüsse am 6510, die sonst ausschließlich für die Bedienung der Kassette gebraucht werden. Dies sind nichts anderes als Stromanschlüsse, die einen HI- oder einen LO-Zustand annehmen, also Spannung führen oder nicht. Programmiert werden diese Anschlüsse mit den Speicherstellen 0 (für die Richtung) und 1 (für den Pegel). Anstatt nun Signale an die Kassette auszusenden (beziehungsweise von ihr zu empfangen), können die Anschlüsse auch zum Umschalten eines EPROMs verwendet werden.

In unserem Fall wird der Portanschluß CASS-SENSE (Bit 4 des Ports, direkt mit dem Kassettenstecker Nr. 6 verbunden) benutzt, der sonst die Aufgabe hat, eine gedrückte Kassettentaste zu erkennen. Da er somit normalerweise als Eingang funktioniert, stört eine Umprogrammierung des Datenwertes (durch fremde Programme) nicht – die zu messende Spannung am Anschluß draußen ist bei Eingängen immer ein HI-Signal. Den Zustand LO (also keine Spannung) kann er nur annehmen, wenn dieses Portbit auf Ausgang programmiert wird, was für die Kassettenfunktion natürlich Unsinn ist (die benötigte Kombination wäre: Speicherstelle 0: Bit 4 gesetzt = Ausgang, und Speicherstelle 1: Bit 4 gesetzt = LO-Pegel).

Genau diese Einstellung wird aber von EX-SMON-Kernel vorgenommen. Die Wirkung ist folgende:

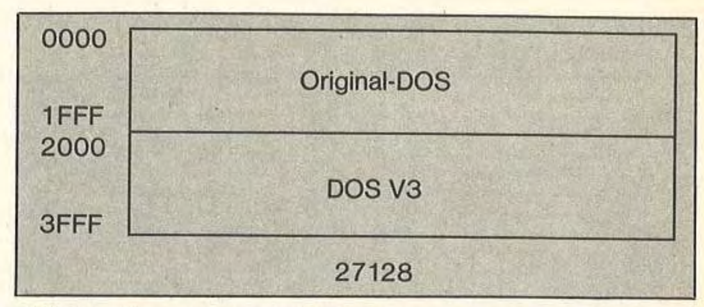


Bild 3. In dieser Form sind das Original- und das neue DOS für die Umschaltung in ein 27128-EPROM zu brennen

Stellen wir uns ein EPROM als eine lange Speicherschachtel vor, dessen Adresse 0 links und Adresse $3FFF (bei 16 KByte) rechts ist.

Das EPROM, dessen höchste Adreßleitung mit CASS-SENSE verbunden wird (der ja eben auf Eingang und somit HI steht), läuft gewöhnlich nur in seinen »rechten« 8 KByte, denn durch den HI-Pegel am EPROM glaubt der Prozessor, die Adresse 0 sei in der Mitte! In diesem »rechten« EPROM-Bereich sind die Originalroutinen des Kernels beziehungsweise 64'er-DOS untergebracht, die den Normalbetrieb des Computers ermöglichen.

Wenn nun eine der neuen Routinen (SMON, PRINT-SCREEN oder COMMAND-Routinen) gefragt ist, so muß genau oben erwähnte Biteinstellung am Port vorgenommen werden, damit die EPROM-Leitung LO-Pegel erhält und dadurch die »linken« 8 KByte angesprochen werden. Speicherstelle 1 ist schon beim Einschalten von der Resetroutine mit dem richtigen Wert (Bit 4 = 1) vorbelegt worden, also muß nur noch die Datenrichtung, Speicherstelle 0, auf Ausgang geschaltet werden, um den gewünschten LO-Pegel zu erhalten. Da der Prozessor daraufhin aber schlagartig auf die »linke« EPROM-Hälfte zugreift, muß er dort unbedingt eine sinnvolle Programmfortsetzung finden. Maschinensprache-Kenner können diese Routinen mit dem eingebauten Monitor gut verfolgen: Der Übergang von der »rechten« Seite zur »linken« liegt bei der Adresse $E47E (LDA #$FF:STA $00:

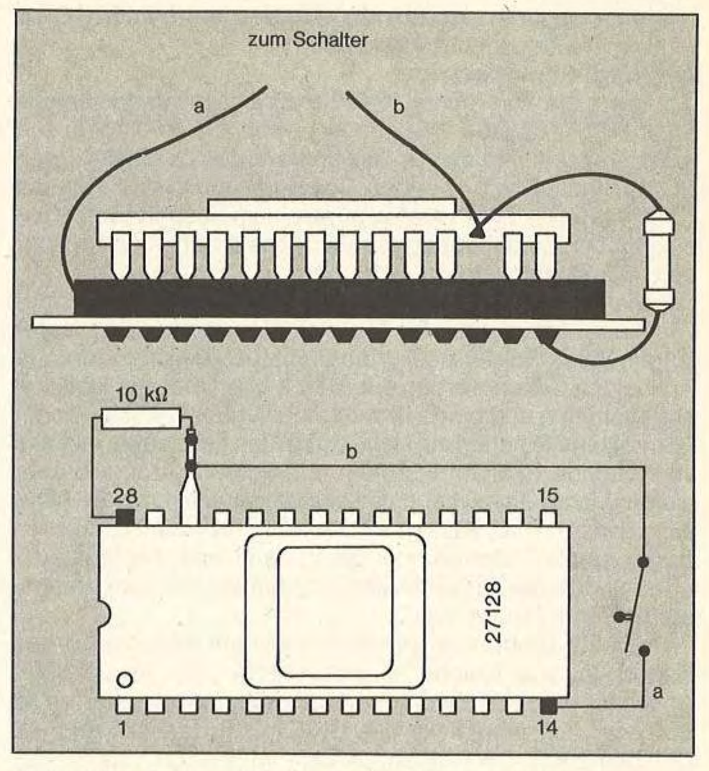


Bild 4. Umschaltung für das Floppy-DOS. Die Zeichnung bezieht sich auf den in Bild 1 erwähnten Adaptersockel

setzt alle Portanschlüsse auf Ausgang). Durch neue Monitor-Registereinstellung kann an der gleichen Adresse (jetzt im »linken« Teil) die Fortsetzung beobachtet werden (NOP als Sicherheit für den korrekten Übergang, dann JMP $E000, wo alle »linken« Programme starten). Durch den Einsatz geeigneter Übergangsroutinen (übrigens alle im Bereich der ehemaligen Betriebssystemmeldung) können die »linken« Programme sogar Subroutinen des »rechten« Bereiches mitbenutzen.

Alles klar?

(Roland Kappeler/dm)

Tabelle 1. Hier noch einmal die gesamten Funktionen des neuen Betriebssystems

Funktionstastenbelegung

<F1> - Directory
<F2> - LOAD
<F3> - RUN
<F4> - MERGE
<F5> - PAGE UP/OLD
<F6> - SAVE"
<F7> - PAGE DOWN/OLD
<F8> - Device #8/#9
<CTRL+F1> - COMMAND-Modus
<CTRL+CBM+F5> - RESET
<CTRL+CBM+F7> - PRINT SCREEN

Editor-Erweiterungen <CTRL+F1>

A - AUTO-NUMBER
D - DELETE
F - FIND
M - MOVE
R - RENUMBER
V - Variablen-DUMP
X - Verlassen des COMMAND-Modus
O - OFF

Monitor-Kommandos

Alle Eingaben erfolgen in der hexadezimalen Schreibweise. In Klammern angegebene Adreßeingaben können entfallen. Der neue SMON benutzt dann sinnvolle, vorgegebene Werte.

Bei allen Ausgabebefehlen ist gleichzeitig die Ausgabe auf einen Drucker möglich. Dazu werden die Befehle geSHIFTet eingegeben.

.A **4000** (Assembler)
symbolischer Assembler (Verarbeitung von Label möglich) Startadresse $4000

.C **4010 4200 4013 4000 4200** (Convert)
in einem Programm, das von $4000 bis $4200 im Speicher steht, soll bei 4010 ein 3-Byte-Befehl eingefügt werden. Dazu wird das Programm ab $4010 bis 4200 auf die neue Adresse $4013 verschoben. Alle absoluten Adressen, die innerhalb des Programmbereichs ($4000 bis $4200) stehen, werden umgerechnet, so daß die Sprungziele stimmen.

.F (Find)
findet Zeichenketten (F), absolute Adressen (FA), relative Sprünge (FR), Tabellen (FT), Zeropage-Adressen (FZ) und Immediate-Befehle (FI)

.G **(4000)** (Go)
startet ein Maschinenprogramm, das bei $4000 im Speicher beginnt

.L **(4000)** (Load)
lädt ein Maschinenprogramm an die richtige oder eine angegebene Adresse ($4000)

.M **4000 (4400)** (Memory-Dump)
gibt den Inhalt des Speichers von $4000 (bis $43FF) in Hex-Byte und ASCII-Code aus. Änderungen sind durch Überschreiben der Hex-Zahlen möglich.

.O **4000 4500 AA** (Occupy)
füllt den Speicherbereich von $4000 bis $4500 mit vorgegebenem Byte ($AA) aus

.R (Register)
zeigt die Registerinhalte und Flags an. Änderungen sind durch Überschreiben möglich.

.S **"TEST" 4000 5000** (Save)
speichert ein Programm von $4000 bis $4FFF unter dem Namen »Test« ab

.T **(4000)** (Trace)
Schrittweises Abarbeiten eines Maschinenprogrammes; Start bei $4000

.V **6000 6200 4000 4100 4200** (Verschieben)
ändert in einem Programm von $4100 bis $41FF alle absoluten Adressen, die sich auf den Bereich von $6000 bis $6200 beziehen, auf einen neuen Bereich, der bei $4000 beginnt

.W **4000 4300 5000** (Write)
verschiebt den Speicherinhalt von $4000 bis $42FF nach $5000 ohne Umrechnung der Adressen (zum Beispiel Tabellen)

.# **49152**
Dezimalzahl umrechnen

.$ **002B**
4stellige Hex-Zahl umrechnen

.% **01101010**
8stellige Binärzahl umrechnen

Der Monitor wird durch ein einfaches <RETURN> verlassen.

```
Name : ker v4 (ex-smon)  0801 433c
-----------------------------------
0801 : 0f 08 ca a8 9e 32 30 36  85
0809 : 35 20 46 43 43 00 00 00  7d
0811 : a0 00 b9 69 07 99 00 cd  26
0819 : b9 69 08 99 00 ce b9 69  ec
0821 : 09 99 00 cf c8 d0 eb 4c  4c
0829 : c2 cd 78 a0 ff 84 fb a9  6b
0831 : c6 85 fc a9 36 85 01 8d  dd
0839 : 20 d0 c8 a5 2d d0 02 c6  97
0841 : 2e c6 2d a6 2e e0 0a d0  a6
0849 : 04 c9 69 f0 0f b1 2d 91  01
0851 : fb a5 fb d0 02 c6 fc c6  10
0859 : fb 4c d3 cd a2 08 a9 01  3c
0861 : 86 2e 85 2d 84 ff 20 50  6f
0869 : ce c9 f3 d0 27 20 50 ce  85
0871 : aa 86 fa c9 04 b0 04 a9  7f
0879 : f3 d0 03 20 50 ce a0 00  97
0881 : 91 2d c8 c6 fa d0 f9 98  03
0889 : 18 65 2d 85 2d 90 02 e6  7d
0891 : 2e 4c 34 ce a0 00 91 2d  77
0899 : e6 2d f0 f3 a9 2d a2 48  ef
08a1 : e4 2e d0 c2 c5 2d d0 be  af
08a9 : a9 37 85 01 a9 fe 8d 20  78
08b1 : d0 58 20 59 a6 4c ae a7  b7
08b9 : a2 ff 86 f7 86 f8 e8 a9  22
08c1 : 01 85 fe a9 7f 85 fd c6  23
08c9 : ff 10 10 e6 fb d0 02 e6  cd
08d1 : fc a9 07 85 ff a0 00 b1  7d
08d9 : fb 85 f9 06 f9 b0 0a a4  6d
08e1 : fe a5 fd 39 f7 00 99 f7  2e
08e9 : 00 8a 0a a8 a5 f7 38 f9  b5
08f1 : e2 ce a5 f8 f9 e3 ce 90  de
08f9 : 0e e0 0c f0 0a e8 38 66  2e
0901 : fd b0 c4 c6 fe f0 bc 8a  e0
0909 : f0 0f a5 f7 38 f9 e0 ce  5e
0911 : 85 f7 a5 f8 f9 e1 ce 85  0f
0919 : f8 a4 fe f0 07 a5 f8 85  ce
0921 : f7 88 84 f8 a5 fd 4a 90  31
0929 : 07 46 f8 66 f7 4c be ce  d9
0931 : bd d2 ce 65 f7 a8 b9 00  63
0939 : cf 60 00 00 00 00 01 06  49
0941 : 0f 2a 65 a9 de f9 fe 00  ae
0949 : 00 00 00 00 00 00 00 00  4a
0951 : 00 10 00 38 00 5c 00 92  69
0959 : 00 cd 00 ef 40 fc a0 ff  ac
0961 : f0 ff 00 00 00 00 00 00  51
0969 : 20 01 d0 85 00 a5 a9 02  c6
0971 : 4c f0 03 60 8d a2 a0 c9  06
0979 : 48 ad dd 90 10 68 0d ff  9a
0981 : 08 29 86 18 04 45 e4 a6  0e
0989 : f3 f1 ea 8a c8 ed b0 ca  d5
0991 : 30 eb 09 b1 2c e6 46 ba  b8
0999 : 84 ae aa 24 e8 bd 06 05  42
09a1 : 8e ee f9 e0 28 49 c6 0a  3d
09a9 : 38 88 ec 40 07 0e fd 5f  00
09b1 : dc e5 a4 b9 91 f7 fb 50  49
09b9 : 14 52 9d f4 fe 13 80 f6  75
09c1 : 0b f2 15 22 44 a8 d3 12  cc
09c9 : 53 69 e9 f5 0c 63 d6 0f  5f
09d1 : ef fc 58 98 fa 11 f8 41  06
09d9 : 4a 42 43 1c 54 16 3a 4d  12
09e1 : 4f 6c d2 e2 59 2b 8c 99  cc
09e9 : c0 4e 64 95 a1 19 3f 7f  7b
09f1 : 97 d1 6d ac b5 cc b7 b8  74
09f9 : c5 e7 21 1e 56 23 2f c4  83
0a01 : d8 e1 2a 2e 81 9a a3 d4  3f
0a09 : 25 65 78 b3 b4 bb 1f 2d  75
0a11 : 51 cd 27 62 7a bf b2 d5  7b
0a19 : e3 1d 39 70 bc d7 17 1a  02
0a21 : 26 47 d9 1b 3b 4b 66 93  93
0a29 : 3d 9e af c2 c3 ce 37 da  3f
0a31 : 94 cb 33 3c 3e a7 ab c7  5f
0a39 : 83 87 8b cf de 57 9b 32  d8
0a41 : 36 6f 73 7b 82 c1 db df  e0
0a49 : 31 55 5a 7c 8f 96 9c 34  d3
0a51 : 5d 67 6b 72 76 79 be 5b  70
0a59 : 89 92 b6 61 71 9f 35 5c  a7
0a61 : 77 7d 6a 6e 5e 7e 74 75  96
0a69 : c0 6f f8 56 7a 9f 54 7f  df
0a71 : af f2 14 a5 3a 51 66 e4  e5
0a79 : af 98 4c ee e2 cb c9 f9  0d
0a81 : a6 3f 99 c2 59 32 67 e6  18
0a89 : 36 21 e0 74 d0 d8 67 8a  9d
0a91 : af ab 41 e7 09 56 ce 75  cd
0a99 : 73 8e 7d f3 19 76 f0 69  0d
0aa1 : 21 bb d8 2e e8 0d 24 1a  57
0aa9 : a2 ba c4 69 20 e1 19 92  a1
0ab1 : a5 41 e5 8a 64 49 50 77  82
0ab9 : a2 57 f2 d0 15 25 72 e8  f4
0ac1 : 04 92 b5 8c 75 34 68 cf  47
0ac9 : 46 67 a3 43 89 47 0a 9e  4c
0ad1 : 6a 72 77 fd d1 af 8d 3b  59
0ad9 : fe e8 d7 a7 ce ab 26 c3  a0
0ae1 : 67 0a a7 e4 fd d4 63 ca  7d
0ae9 : fe 94 7c d8 9e 77 87 9a  64
0af1 : f0 ff f3 9d 63 46 ac b2  12
0af9 : d1 03 34 f4 de cd 45 8e  86
0b01 : ca 68 d5 16 32 80 7d 52  f9
0b09 : bc 45 a2 06 69 ea 1d 37  a2
0b11 : b2 bc 23 7b 0e 37 93 3f  c1
0b19 : 21 eb 16 63 e6 0f aa a9  07
0b21 : 58 4b 55 06 69 e8 4d 48  d9
0b29 : 27 09 1a 20 97 90 09 3d  fc
```

Listing 1. Die zum Brennen benötigte Software für das EX-SMON-KERNEL. Bitte mit dem MSE eingeben

```
0b31 : 55 bb 1f 72 8d 16 f1 99  fe
0b39 : e8 d5 d1 21 1b c3 85 4f  29
0b41 : 35 39 3b d4 a3 5f 1a 77  09
0b49 : a9 46 bd 3d 26 24 b6 b2  f0
0b51 : e7 5b 9f e4 e6 fa 79 54  3f
0b59 : b9 54 90 76 fc 6f 01 ee  5c
0b61 : af c3 3e c7 5f 07 5f 05  30
0b69 : 56 d3 8e 62 d1 ee 87 4f  ea
0b71 : 2a 65 85 b3 ad 12 1c 8d  1d
0b79 : 87 07 cd 06 be 1c 59 c0  6c
0b81 : fd c7 05 eb e7 d1 59 7c  8c
0b89 : ea e8 9c c5 57 aa e5 47  b8
0b91 : d5 ee 64 cc 99 3c 1e 2c  dc
0b99 : a0 dd bd d9 23 de 54 07  5b
0ba1 : 17 14 89 d3 57 ac 7b a0  a9
0ba9 : a8 7c f7 c1 1b be 2c 9a  53
0bb1 : 38 6e 2f 87 99 4c a0 f6  49
0bb9 : 14 d1 55 ef 5c a8 3f 69  e4
0bc1 : 71 cc 9b ae 99 65 54 bb  e2
0bc9 : dc 8b 38 18 0f 50 c3 e7  ce
0bd1 : 54 c9 d7 8f d7 95 af 19  0d
0bd9 : f2 a9 d3 45 66 60 8d 61  1f
0be1 : 0f 5f c3 58 5b 3a d1 49  fd
0be9 : ec 91 6a fb 37 39 19 9b  91
0bf1 : 55 77 5e c5 68 99 df d0  c7
0bf9 : d6 89 0e 46 c3 83 e6 b3  3c
0c01 : aa 06 be 1c 5e 30 7e e3  0b
0c09 : 82 f5 f2 c5 d9 80 a1 c9  b7
0c11 : d7 a3 df ab a2 73 15 5e  fe
0c19 : ab 95 1d 2c 20 d8 2c be  12
0c21 : 10 d2 43 c5 d8 2f 59 28  e1
0c29 : 72 75 e8 fe b5 d8 63 a7  6f
0c31 : 60 02 a5 6c a0 08 ec 39  fa
0c39 : 40 14 74 a6 a3 e6 53 28  84
0c41 : 3e 75 34 55 7b d7 2a 0f  2f
0c49 : cd 5c 73 26 eb a6 59 55  ea
0c51 : 2e f7 22 ce 06 03 d4 3c  22
0c59 : 7e 75 34 58 f5 a7 15 40  1b
0c61 : 13 42 26 d0 95 e8 62 f3  4b
0c69 : 9a 34 58 f9 3a f5 f0 e4  53
0c71 : eb d7 85 eb e6 84 c2 68  95
0c79 : 4b 84 32 79 cb 25 27 b2  aa
0c81 : 26 15 4c 9e c8 97 15 32  1b
0c89 : 7b 24 52 43 7b ce 0f 5a  32
0c91 : 43 29 c4 12 b1 68 73 65  d3
0c99 : e3 d7 a8 5b 06 0d 6c 0d  92
0ca1 : d2 5c 90 f0 d8 68 86 e1  92
0ca9 : 31 f9 cc 3c a1 b8 10 ee  8f
0cb1 : 18 6a 86 e1 98 da b0 2b  56
0cb9 : 17 09 2e 12 35 8b 92 45  a7
0cc1 : 24 3a 6d a4 9d a8 56 2c  c3
0cc9 : 1f 1e e0 07 01 02 36 24  52
0cd1 : 99 c5 24 52 32 39 d8 7b  e7
0cd9 : 52 0c c8 7e 33 07 43 71  8f
0ce1 : 9c 5d 3e 8d 62 ea c1 e6  bf
0ce9 : 8e fd b2 87 26 1b 09 06  7f
0cf1 : 64 3f a0 e9 c2 78 27 24  2f
0cf9 : 4e c2 c5 39 a8 41 91 36  88
0d01 : ca 3e 22 b4 e7 00 41 91  b0
0d09 : 36 ca 3f 8b 18 77 ea d2  74
0d11 : 49 c5 22 8e 37 73 ca 4d  6c
0d19 : a8 cc f9 2b ef 18 f9 2a  07
0d21 : b0 83 22 6d 94 7f 3a 56  a4
0d29 : 28 cf 95 93 a6 f0 46 ee  f9
0d31 : 9b b1 1e 7e 30 f3 68 f6  2e
0d39 : 8e 6a 10 64 06 0c 84 37  ce
0d41 : ad a0 c1 3b 3c a7 54 48  f9
0d49 : 7e c7 be 3c d7 87 31 df  20
0d51 : b8 49 ad 49 e4 fb 90 fe  b1
0d59 : 25 56 e5 67 a1 9a 7a fa  de
0d61 : 5e 8d bc 53 d5 6f 0a 94  49
0d69 : 44 97 92 8e 12 f2 01 27  fa
0d71 : a8 25 62 b2 56 77 ed b6  e1
0d79 : 8e d2 a9 67 6a 77 a2 a6  02
0d81 : a9 23 af 14 f2 52 58 5b  04
0d89 : 37 c7 9d cb 53 ce 12 bf  f8
0d91 : a5 6a a0 cd 3d 41 2b 10  f8
0d99 : f5 a4 1b 64 48 a6 94 24  88
0da1 : 99 bb 95 19 47 93 64 6e  20
0da9 : 86 51 49 28 c0 f4 89 e0  cb
0db1 : b0 49 22 76 0e a1 50 f4  76
0db9 : 8e 45 44 92 7d 9b 59 d0  09
0dc1 : 6b e4 2b ea 2c f4 33 4f  9c
0dc9 : 5f 66 d6 28 6b e4 53 19  73
0dd1 : 0f 05 e9 69 2d fa eb 43  eb
0dd9 : 45 d5 5b 46 5c 4a 36 f1  7d
0de1 : f3 f7 d8 a3 d1 82 ff 1b  e2
0de9 : 41 b5 cf aa b9 67 c3 c0  b5
0df1 : 84 ab 67 3a b5 7e c5 b2  38
0df9 : c6 52 49 1d a1 53 00 46  20
0e01 : 92 a0 9b 16 cf aa 29 9f  c3
0e09 : 2c 54 05 4a 67 bd 15 00  a3
0e11 : 92 43 f6 3d 56 3a dc ea  2b
0e19 : 09 b1 6c fa a2 97 a0 c0  60
0e21 : 10 ff 32 f3 9a 84 bc 86  0a
0e29 : 89 99 4a 8c a9 f2 29 16  a6
0e31 : 7a 9a 2e 4f 7a 45 13 f7  7c
0e39 : e0 46 81 d4 c1 c4 b1 cc  da
0e41 : b1 cc 89 67 11 8c 4b c8  dc
0e49 : 9e b4 ef 79 3a 99 b9 1f  02
0e51 : 0d ec 1b da 1d cd b7 d1  b9
0e59 : 03 4a fc 56 98 36 f8 26  f7
0e61 : bb f7 5f 89 ae 38 4b 52  9f
0e69 : ed b1 03 fa ea ca d7 11  d5
0e71 : e7 26 97 4e f0 39 40 5e  b2
0e79 : 07 9c 29 82 c1 73 b7 98  31
0e81 : eb 2c 59 3d ba 37 41 2f  49
0e89 : 22 18 ef da e3 85 2e db  e9
0e91 : 21 e5 ce a4 1a b5 29 60  a2
0e99 : c1 40 b4 67 67 48 58 21  f1
0ea1 : ac b1 0f 5a 40 6a ac 43  c6
0ea9 : 59 62 82 76 30 5a 52 27  10
0eb1 : 29 51 ab 02 e7 d5 c4 59  a1
0eb9 : ea 1f 3f 87 9f 57 62 b5  9d
0ec1 : f8 ef f3 ea f4 56 80 3c  87
0ec9 : ee 7d 5d 32 d7 e3 9c 38  93
0ed1 : 4a 56 89 5c 55 aa 83 34  55
0ed9 : f5 57 4b 44 0c d3 d0 e1  3c
0ee1 : 44 7d 71 41 9c 11 b0 94  a7
0ee9 : 66 25 7e 65 aa 83 34 f4  b0
0ef1 : 38 51 01 27 a6 c2 1e 7d  eb
0ef9 : f5 c2 3a 9a 0b d1 07 ca  22
0f01 : f7 d8 ad 12 2d c1 27 aa  e5
0f09 : df 21 1b c3 0d 65 88 27  25
0f11 : 62 1a cb 10 d5 58 a3 05  2e
0f19 : 85 68 87 80 b1 e6 cd 5a  02
0f21 : 57 c3 37 7d 69 5e b1 64  f0
0f29 : eb e6 a1 c9 d7 b9 af 22  77
0f31 : 63 12 66 cd 3a df de 88  20
0f39 : 99 a6 5e 41 bb ec 92 93  7a
0f41 : be 5d 6c cb bc 56 b4 7c  8d
0f49 : 43 6c 89 5e 69 42 26 c3  b9
0f51 : 38 22 66 98 33 11 3b e4  b9
0f59 : d6 3d f6 6c 35 f2 16 3e  d9
0f61 : 4e bc 9e 2c d4 39 3a f2  20
0f69 : 6a f8 15 a2 1b 50 3e a9  69
0f71 : 67 67 00 4b c8 a9 1e 84  51
0f79 : 20 4a ac 55 67 a4 eb db  97
0f81 : ca d7 8c f9 54 e9 11 c3  f9
0f89 : 5d 86 3a 76 00 2a 51 28  6e
0f91 : 02 3b 0f 80 02 1b 1e 07  84
0f99 : 1c 43 e6 b1 9f e3 23 81  ef
0fa1 : 19 9e 8e 3f b1 0e 4e bc  d3
0fa9 : 7b 94 e8 e7 87 58 73 65  79
0fb1 : 38 e2 1c ed 40 71 71 68  45
0fb9 : c4 8e 05 1a 4c b0 83 ed  7d
0fc1 : 2c 5e 03 29 b0 fb 4a 6a  eb
0fc9 : f7 59 27 92 bd 91 f2 29  10
0fd1 : 87 da 56 5a 25 7f 12 d5  e8
0fd9 : 41 9a 7a 1a 19 07 36 52  91
0fe1 : b4 43 6e 1f b8 e0 f2 e5  e0
0fe9 : 59 71 aa 03 8b 8a 1f 69  62
0ff1 : 59 5b 2e 94 1f 82 e8 dd  7b
0ff9 : fe 48 89 5e 6f bb 24 a4  f8
1001 : 4c 74 c6 af 5e 64 d7 87  a6
1009 : 8b c6 8b 58 d0 89 b4 25  5c
1011 : 7a 6c 2b 44 3b b0 d6 58  5a
1019 : ac 93 a0 a8 7d c5 b8 1b  eb
1021 : b7 b8 48 f9 4b 16 4e be  a2
1029 : 6a 1c 9d 79 13 73 5e e8  50
1031 : d9 ad fd e8 89 9a 65 61  43
1039 : 2b df 0f 9a ab 2a 80 a5  c4
1041 : f1 c1 8c eb 42 26 4f b9  b9
1049 : a1 2b c9 df ac 59 3e fd  79
1051 : 70 23 80 bb 57 16 9f 2b  e5
1059 : 97 51 94 38 a9 43 1d 54  97
1061 : c5 02 a5 9d 10 91 4a 70  dc
1069 : 7a 3a 0d 40 a1 54 58 d0  0b
1071 : 29 e2 09 1f 66 28 1a 82  47
1079 : a2 26 a0 4a 78 82 74 6a  e2
1081 : f2 4d ef fc 41 13 0c c9  26
1089 : 93 ab 92 52 72 83 1f 8c  ba
1091 : 3c 27 46 ee 13 f3 6f fc  59
1099 : 41 13 3e 6e 81 3f 9e 49  e0
10a1 : 49 b1 0f 03 a6 b1 6c c4  1a
10a9 : 24 67 ed 2b e1 8e fc 3d  63
10b1 : 69 15 09 ba 35 7e 4b b2  18
10b9 : 57 b2 3e 55 33 8a a7 1c  02
10c1 : 4a 00 93 53 dc fe 90 be  e0
10c9 : 0f a7 f4 18 4e 2b 88 25  97
10d1 : 8a 6e 57 17 a0 39 e3 61  71
10d9 : f1 7b 88 e3 c9 e1 da 4d  d8
10e1 : ca e1 8c f9 5e 54 ac d8  eb
10e9 : e5 65 7b a3 d8 18 52 47  ba
10f1 : 90 49 9b 64 fa 51 61 12  7d
10f9 : bd ca 72 75 e8 ea 21 c3  58
1101 : 36 16 fa bd 14 aa 47 58  1d
1109 : 6b 8a 9d f5 7a 38 15 32  01
1111 : 75 f3 55 32 75 e8 7a 7c  9d
1119 : 67 49 9c 55 30 c4 b0 a4  2c
1121 : f6 43 9f 56 f9 08 de 1a  fb
1129 : b4 2b 55 06 69 ea bc 35  d4
1131 : aa 83 34 f4 38 51 02 f5  4a
1139 : f1 ab d4 d1 09 4d 50 8c  c4
1141 : 9a 98 e2 53 c9 12 be 02  76
1149 : d5 41 9a 7a 1c 28 92 b7  71
1151 : 03 3e 57 08 2f 5f 3e 4d  cc
1159 : 86 88 55 58 e2 46 7e 2d  38
1161 : 62 14 b6 3c 5a f8 b4 4e  df
1169 : 76 66 af 27 cb 51 13 44  ff
1171 : 25 1d 0a 14 4c 71 29 ba  94
1179 : 12 bf 12 d5 41 9a 7a 1c  b5
1181 : 28 92 b7 03 3e 57 08 2f  5e
1189 : 5f 3e 4c d5 f4 18 42 52  93
1191 : e0 4e 7b 04 f0 55 15 05  10
1199 : 68 a8 2f 45 1a 83 38 2a  bd
11a1 : 06 62 0f 95 ec a5 68 a3  34
11a9 : f2 1f 76 0d e0 d5 79 22  51
11b1 : 65 e7 0e b3 bc d5 09 95  ce
11b9 : c2 39 8f 76 6c b7 a3 92  03
11c1 : e2 fc ac ad 06 92 1e 37  de
11c9 : 1b d8 c3 11 ba 0b cd dc  58
11d1 : 5e a9 68 39 42 99 dd 0a  c2
11d9 : 8c 22 ee a5 30 c4 cf 12  73
11e1 : a1 85 c0 a6 70 85 11 d0  63
11e9 : 89 e0 c2 89 ba 11 3b 03  eb
11f1 : 8f ec 72 84 a9 68 1d 08  86
11f9 : e0 53 74 29 4c 31 47 d1  d4
1201 : 3a df e1 09 4c 21 30 c7  e2
1209 : 16 48 29 8e 21 b5 36 ce  96
1211 : a3 28 09 b3 37 63 28 09  c2
1219 : d8 54 b3 b9 9c 6b aa 41  92
1221 : 2e 4d 14 a7 06 fc 4c 1d  a4
1229 : 08 99 f6 1b a1 3a c9 a3  79
1231 : 26 c5 86 a8 44 af 75 bf  08
1239 : 74 22 7e 66 1e 48 88 9e  ae
1241 : ff 74 22 7f 30 f0 3a 6c  3f
1249 : 35 42 24 db ad fb a1 13  8b
1251 : 0d 87 92 22 75 77 fb a1  51
1259 : 13 94 14 82 34 d9 3f 31  7d
1261 : ba a4 fe 66 f0 49 5e 6e  a9
1269 : c1 13 37 70 99 f3 74 09  ad
1271 : b1 0f 03 a6 71 49 86 c1  3e
1279 : d0 89 dc 3a 8c 9c a7 5b  8f
1281 : fd d0 89 d0 37 82 49 b3  77
1289 : 76 09 d5 0a 41 34 27 71  70
1291 : a1 3a 12 9c a1 33 e7 b0  1c
1299 : b8 14 ce 10 89 e0 b9 ec  71
12a1 : 4c e3 89 86 88 4a 6a 85  a2
12a9 : 41 e4 88 df 2d 44 4d 10  c5
12b1 : 94 74 23 6c e2 63 89 4d  e0
12b9 : d0 90 e5 ce 07 ad 21 0d  a1
12c1 : ab d5 68 96 aa 0c d3 d5  4a
12c9 : 78 6b 55 06 69 e8 70 a2  f2
12d1 : 05 eb e3 57 92 6c c9 93  8a
12d9 : aa 1e b4 85 7c 05 aa 83  12
12e1 : 34 f4 38 51 25 6e 06 7c  9e
12e9 : ae 10 5e be 37 57 8a 9b  9e
12f1 : 2c 04 52 42 96 c4 d3 7c  d4
12f9 : 5a f8 b6 a6 af 26 19 93  09
1301 : 27 28 31 f8 cb 0a 4f 64  bb
1309 : 8a 41 eb 48 41 2b 15 67  c8
1311 : 96 88 19 a7 aa fc cb 55  93
1319 : 06 69 e8 70 a2 1b 09 14  6b
1321 : 90 a5 b1 35 c2 34 4a 5e  4b
1329 : a4 c1 f1 f2 55 6a 3e 3d  a5
1331 : 47 c7 a0 95 8b 45 eb b7  39
1339 : 23 d5 54 c9 f7 e3 b8 60  d7
1341 : f8 f7 1e 02 8f 8f 43 e1  43
1349 : 7a d0 97 0d 09 84 87 c2  0b
1351 : f4 d8 53 4b cd 70 bd d6  f5
1359 : fe f5 26 42 9c 26 68 97  ef
1361 : 0e 7b e7 38 4a b2 a8 0a  1f
1369 : 5f 1c 18 ce aa 29 d2 8a  0b
1371 : eb f8 1e 8a 53 83 cb 9c  6b
1379 : ba fe 07 9c 17 5f c0 f7  67
1381 : 34 c3 fe 4a f9 8b 3d 0c  a9
1389 : d3 d5 00 4a 71 c4 14 f5  0a
1391 : 4e 7b f9 8f b8 3e d2 cd  71
1399 : 0d 24 3b c8 2b b0 c6 95  1f
13a1 : 0f 1c 41 f9 cc 09 3d 0f  76
13a9 : 5a 41 b6 44 af 34 a1 13  23
13b1 : 58 f9 3a f2 29 0e 4e bc  a8
13b9 : 94 10 e4 eb d1 41 76 18  3d
13c1 : e9 d8 00 a9 44 a0 08 ec  8f
13c9 : 39 90 15 57 1c 43 b1 62  e2
13d1 : 53 c7 47 39 0e 74 a7 a3  6b
13d9 : 3d be af 1e 6d 17 e2 d0  e2
13e1 : 2f 35 03 e5 00 fa a3 63  56
13e9 : 54 65 53 c5 cf 52 b6 c9  7b
13f1 : 14 d4 e4 64 6b 0e 00 97  8b
13f9 : 90 3b f6 98 a3 8a b1 76  3a
1401 : 60 29 3a f9 a8 72 75 ee  95
1409 : 6b c8 9c 9d 78 ed 01 27  fc
1411 : a8 25 62 af d6 b4 40 cd  8a
1419 : 3d 37 72 cb a0 4b 89 8c  ab
1421 : 2e 64 71 9f c5 e5 b2 ac  81
1429 : 57 ff cb 59 21 f3 f8 79  26
1431 : 27 b2 14 32 61 cc 72 82  48
1439 : 6e 40 7e 59 10 99 cf 06  ab
1441 : fe 42 18 52 34 11 e7 83  23
1449 : fe a4 21 85 20 1c c8 04  a0
1451 : 9e 9a e2 cb 09 d6 4d 63  b2
1459 : 29 d8 71 65 d6 2c 9d c4  c6
1461 : d3 8d e0 4b c8 87 27 71  e5
1469 : 8a e3 78 8a 41 2f 22 69  3d
1471 : 9e 8b 41 1e 9c ee 38 4b  a2
```

```
1479 : c8 68 d1 0c 29 07 75 47  9a
1481 : f9 b4 0c dd 9d 36 3c 84  19
1489 : dc a8 fb 37 31 40 b3 b0  e4
1491 : d7 c8 7c a5 95 01 de 98  ae
1499 : 4b c8 38 07 ab 58 b2 78  71
14a1 : 73 50 e4 f0 f1 61 06 e3  9d
14a9 : b0 85 f4 80 7d da 06 33  96
14b1 : a8 73 ea 80 d2 16 3e 4f  53
14b9 : 0c 9d 5a 99 3c 32 4d d4  92
14c1 : c9 e1 93 cb 46 9e 4e 6c  45
14c9 : 25 e4 43 e6 79 63 bf 3b  36
14d1 : 0d 09 79 0d c9 24 d1 c5  f3
14d9 : b2 f3 49 2f 84 bd 52 a3  84
14e1 : ff ad d7 c2 5e a9 e3 e4  91
14e9 : b7 5f 09 7a ae c7 b6 6e  c2
14f1 : be 12 f5 67 87 2c dd 7c  6d
14f9 : 25 ea b7 1f be dd 7c 25  7c
1501 : ea d3 8f eb 6e be 12 f5  47
1509 : 73 07 d0 6e be 12 f5 6b  2d
1511 : 47 e5 b7 5f 09 7a ac c7  cb
1519 : c8 6e be 12 f5 71 c7 aa  6a
1521 : 6e be 12 f5 6e 07 5a dc  74
1529 : 92 4d 1c 5b 2f 34 92 f8  a5
1531 : 4b d5 d5 4e f9 cf 54 9b  4d
1539 : 14 e4 ef 9c f5 4e a8 9a  f8
1541 : 4d 20 e5 eb 75 f0 97 aa  28
1549 : e8 72 ad d7 c2 5e af 94  d8
1551 : 3f 9d ba f8 4b d5 d4 0f  01
1559 : 36 dc 92 4d 1c 5b 2f 34  0d
1561 : 92 f8 4b d5 e4 0e 99 b9  95
1569 : 24 9a 38 b6 5e 69 25 f0  67
1571 : 97 aa 03 fe cd d7 c2 5e  61
1579 : ae 18 e6 5b af 84 bd 59  21
1581 : b1 be 6e be 12 f5 7c b1  2b
1589 : fa 8d d7 c2 5e ad 20 ff  6c
1591 : 4b 75 f0 97 ab cf 1d 03  7a
1599 : 5d 3e b7 59 ac d2 6d b1  a9
15a1 : 03 76 05 6a f3 3f a9 e8  20
15a9 : 64 6c 72 35 2f 35 81 ad  85
15b1 : 96 96 e8 35 2b 5c 70 b9  3e
15b9 : 79 b6 fc ba cd 33 cb 90  ea
15c1 : 6a d4 b2 d7 3a dd 60 6a  26
15c9 : c0 ba 9d 68 00 6a c0 bf  31
15d1 : 42 f7 97 2f 35 60 5c 8d  bd
15d9 : 4b cb 70 d5 87 e4 8f 75  a9
15e1 : 74 56 bf 97 6d 88 f2 5b  01
15e9 : bd c4 0c 8d 4e 93 52 f3  70
15f1 : 58 f3 6d a9 2b 5c 70 d6  d8
15f9 : 6b 79 2f 01 ab 52 b5 c7  c1
1601 : 0c 47 88 3c d4 bc 06 ad  01
1609 : 4a d7 1c 34 9f 97 53 ad  2c
1611 : ee 6d bf 2e b3 4c f2 e4  9f
1619 : 1a b5 2e 46 a7 49 73 a9  48
1621 : d5 87 e4 8f 1e ea ea e8  9c
1629 : ba 4d 48 00 6a c0 a8 52  90
1631 : ed 9e 64 6a c0 be 96 38  a0
1639 : 72 74 97 34 ba 4d 32 0f  4f
1641 : 32 01 ab 52 fa 4d 5f bb  38
1649 : 07 ee f9 e1 a7 d6 eb 35  cd
1651 : 9a dc 4d 6d cb c0 fa ff  09
1659 : 58 69 b4 99 14 ba 70 e4  69
1661 : fd 40 fd df 3d ab 57 f3  70
1669 : c1 f5 fe be f5 e5 c8 62  33
1671 : 6b 72 03 64 0f 33 fa 97  08
1679 : 9b 6c 47 98 80 f3 12 97  4e
1681 : 53 ad b9 6a fe b0 3e aa  2a
1689 : 8f 40 2e 75 ba c0 d9 07  9a
1691 : d7 fa fd 00 b9 d2 6b 03  4b
1699 : 23 63 91 a9 79 ac 6a fd  aa
16a1 : b8 14 34 da 4c 8a 5d 38  cb
16a9 : 69 b6 3a 97 99 01 2d 72  2a
16b1 : f1 e3 56 a5 0a 7a a7 3d  6c
16b9 : fc c7 dc 1f 69 50 fd c7  54
16c1 : 2c 94 c6 18 6a ea 89 ca  a6
16c9 : 0d 5d 55 93 21 fb f4 d8  c4
16d1 : 63 45 f7 28 34 d5 47 00  e9
16d9 : 4b c8 0d 5d 53 e6 43 70  d2
16e1 : fb 1b 14 6e e8 fc a8 b8  c7
16e9 : 4b 9a a2 ed 72 83 3c ee  fa
16f1 : d8 4b c8 39 22 75 09 cd  56
16f9 : 1f e2 39 a8 4b c8 58 f9  3d
1701 : bb 48 fd 94 2c de 97 77  f9
1709 : ca 9b ea ca 6a fc fd 26  88
1711 : 9f 12 5b cc 43 59 ad b9  53
1719 : c8 d4 f2 7e 82 97 c0 bf  3f
1721 : 96 b5 0b 5f d7 6a d7 2f  cf
1729 : 3d 03 f4 40 d8 5d e8 ed  25
1731 : 7d f7 e2 fd fa 3f 02 06  e0
1739 : c6 ff 37 02 62 d7 fd c0  6b
1741 : d8 df ca 68 7c ad 1e f6  64
1749 : d5 b3 61 17 49 f9 a9 6e  1b
1751 : 75 37 c1 56 cc a7 88 47  58
1759 : 95 ab 50 00 b7 d3 80 64  bd
1761 : 5c 86 93 4e 19 fb 90 d8  14
1769 : dc 86 45 b8 69 ed c0 36  66
1771 : dc 9e e6 9b 12 97 f3 69  4a
1779 : ca 7b b1 4a 16 53 62 e3  04
1781 : 3f 75 db 6d a8 d4 76 fb  22
1789 : 7d bd 47 e3 d4 76 ff 1b  6a
1791 : 66 c9 76 ef a4 83 31 db  5a
1799 : 31 68 ae 3f 4e a3 75 ed  46
17a1 : ff 2d 72 57 a2 3c f4 92  c3
17a9 : 7f 3c 7a 49 1e 92 49 1e  e6
17b1 : 5a ba bb eb e0 67 d5 f2  5b
17b9 : f8 f8 7d 1f bf 8f f7 cc  62
17c1 : 37 6e fd df 19 fd 3f 87  38
17c9 : c6 f8 38 dc 67 e7 da fe  d4
17d1 : 3f 2e 5a 7b d1 d2 74 79  a6
17d9 : c2 59 5d cf e1 c7 fe f3  d9
17e1 : 0e 65 69 a8 f6 b3 68 03  c6
17e9 : 28 a8 c7 1b 3b 7f 1b ae  34
17f1 : 57 5c dc ff 12 35 4c e4  73
17f9 : 38 2b db b4 a4 ab 4e 6b  0c
1801 : 16 5e 17 4b 41 51 a3 fe  a1
1809 : 9f ab d4 f0 79 fa 5a 7f  a9
1811 : fb d6 f2 bd 2e c3 f1 7b  ab
1819 : b8 d7 b9 1f 6d ca 1a 4b  3b
1821 : 97 99 19 1a d4 10 43 4f  88
1829 : a4 d2 64 64 52 e2 69 f4  a8
1831 : fa 6d 2e 95 fb 7d 36 46  31
1839 : 99 fb bd 3e 26 25 2d 2e  a4
1841 : d9 fb bb 7b 9b 97 f3 da  72
1849 : 9c 8c 8d 6e 46 46 b7 5b  89
1851 : a9 d3 5c e2 62 62 6b 36  ab
1859 : d8 9a 9d 4e d9 e5 bd bd  ef
1861 : 2e b3 23 4f a7 79 ac db  cc
1869 : 5b f2 79 2f 2e 5f bb 41  d1
1871 : e3 cd b6 db 5a bd 7a 08  f1
1879 : 6a 75 3a 4d 26 46 7f 3f  e7
1881 : b1 78 f1 0b 94 2e 6e 74  a9
1889 : 9a 7d 26 c7 3f a4 b7 c4  e6
1891 : d3 ec 73 fa 7d 3e 7f 63  a5
1899 : a9 b7 b9 d3 e4 77 69 5e  73
18a1 : 21 a7 c8 a5 f4 34 f8 94  7b
18a9 : bc 9c fe c7 3f b1 b7 d2  72
18b1 : 5b e9 2d e9 72 33 fb 1c  72
18b9 : fe 93 23 49 a7 c4 6f 31  34
18c1 : d1 c3 e3 f9 61 7c 90 6a  bd
18c9 : 2e 2e 19 bb cc e6 9c dd  fe
18d1 : 48 bc d5 6d a8 30 05 41  3d
18d9 : 8e 2e 93 ec dc 65 8a 1a  58
18e1 : f9 0a fb 21 df b6 83 05  ce
18e9 : 0a e4 79 aa cf b4 ba cc  40
18f1 : 14 66 74 4c 1a 88 69 7b  61
18f9 : 19 82 f6 1a 8a 5e 83 1c  36
1901 : 4a db 50 60 09 f0 a8 2f  72
1909 : 44 75 ca 0e e0 04 64 51  df
1911 : c0 07 28 a8 d4 46 96 53  f4
1919 : 30 49 ab e2 85 4a 21 52  09
1921 : 88 f5 00 e5 13 af 09 e6  01
1929 : 1c d8 8b c2 32 23 cc 53  03
1931 : fe 1f 2b 4c 98 39 46 51  22
1939 : 57 81 b8 47 70 20 55 1d  ff
1941 : c1 67 27 13 0d 7a 2b 81  36
1949 : 1a 25 d2 88 e5 8d 20 d7  b6
1951 : a2 1c 34 5f bb a7 25 28  d8
1959 : 6b d1 0e 1a 33 f4 a9 ae  53
1961 : d5 b2 3f 25 bb 8c e2 61  72
1969 : af 45 70 23 45 4d 8c 98  5d
1971 : 5c d5 53 63 2e 6a 98 5c  4a
1979 : d5 2e 6a 3d 1a 09 58 9d  2e
1981 : bb 09 79 13 38 e2 e9 2a  18
1989 : 57 34 c0 93 dd ce f9 53  80
1991 : d8 f2 e8 51 6f 19 9e 8c  9a
1999 : ee 80 4b c8 0d 7a 2b ba  7a
19a1 : 08 d1 13 48 e3 78 ae 13  43
19a9 : 1e 7d 0a 33 f1 87 a3 46  e5
19b1 : 18 79 50 10 e7 23 0a 9e  99
19b9 : 6a 72 77 77 42 be 34 ee  f2
19c1 : ee 85 7a 66 96 0e 5c 03  2f
19c9 : 3c 5e fd de 94 35 70 1a  78
19d1 : 38 ce 2e a7 64 7e 4c f6  4a
19d9 : 4c 74 25 e4 1c d4 25 e4  0c
19e1 : 42 42 5e 40 77 60 1c 08  df
19e9 : 4b c8 95 78 33 e5 65 93  2c
19f1 : 5a 85 5f ce 19 a7 aa f9  2d
19f9 : e0 f2 00 77 60 07 dd 00  f7
1a01 : 3b b0 03 a3 01 85 68 90  c9
1a09 : 61 00 35 34 48 52 40 57  05
1a11 : 43 3b 70 50 e5 c0 0e ec  8e
1a19 : 09 c1 dd a6 1c 94 55 d0  ac
1a21 : 86 bd 10 55 1b ab 82 ce  eb
1a29 : 0e ee 67 1c 54 18 02 0e  36
1a31 : ec 06 15 a2 26 ec e0 5a  bc
1a39 : 48 25 e4 4f 6e ce 8f 92  f8
1a41 : 5a 91 dd a6 1d d4 4c f0  3c
1a49 : 97 06 96 26 12 ba 1e 91  e0
1a51 : d6 4b ce fb 57 a0 fc 5e  2b
1a59 : 18 6b 91 57 56 0f 20 05  df
1a61 : 24 05 81 52 87 76 00 35  49
1a69 : d0 87 2e 00 35 d0 83 ac  ca
1a71 : 11 e7 43 d9 c1 65 f3 a6  e6
1a79 : 0e 51 58 15 28 6b d1 05  18
1a81 : 51 0f 9e 89 14 90 b0 44  44
1a89 : 1d 60 8e 22 61 ae bb 6e  16
1a91 : bb a0 8b 7a 46 f1 4f 71  e3
1a99 : bb 71 bc b4 df 30 72 7d  17
1aa1 : a3 f1 34 0e 4c 08 7b 5a  b3
1aa9 : 44 c2 97 05 0a bc 10 e4  65
1ab1 : c0 33 ce 2f 98 69 6b 15  51
1ab9 : e9 86 ff 04 13 c1 46 b4  27
1ac1 : 13 c1 46 d9 70 0e 7d 4f  8e
1ac9 : f8 7c ad 32 61 35 21 7c  ee
1ad1 : dd a3 33 da 26 87 39 15  56
1ad9 : a9 26 f4 8c ef e6 21 cc  b8
1ae1 : 0c e2 91 91 ce c3 da 90  8c
1ae9 : 66 43 f1 0e 4c 36 12 26  3a
1af1 : d9 47 b8 54 ae 71 a0 c1  a3
1af9 : 0b 9a a0 83 22 6d 94 77  b9
1b01 : 6b 9a 8f 04 30 64 26 fd  d8
1b09 : dc 15 8a 1b 7a 29 bd dc  17
1b11 : 73 ed f3 15 ed b1 8f 9d  00
1b19 : 46 3c 6a 3c 21 df aa ed  37
1b21 : 8f a7 9d 47 0d 8e 93 3a  dc
1b29 : 8f 6e 30 df 89 5a 86 33  e3
1b31 : f9 d4 70 a3 51 ea 87 7e  ac
1b39 : 1b 7a 2f 4f 3b 43 fb 27  53
1b41 : f3 a8 cf c6 1e 0a a5 73  05
1b49 : a6 84 8b 35 74 41 b7 a2  30
1b51 : f4 ce a1 fd 92 f4 28 a5  91
1b59 : 18 78 33 bf 3d 09 2a e7  07
1b61 : 78 54 24 f4 1f 69 93 58  e7
1b69 : 41 b7 a2 9a 86 1b 7a 25  f7
1b71 : 99 f3 2a c7 58 1b 27 e2  48
1b79 : 59 21 9f 65 e8 39 d3 93  c6
1b81 : 11 35 84 1b 7a 26 62 a9  67
1b89 : a8 eb 3c 9a 9c 44 55 3a  3f
1b91 : 0e b4 b2 6a 95 ce ec 35  e1
1b99 : c8 86 fd f8 8d a6 5d aa  1c
1ba1 : 45 ba e8 e8 bb 5d c7 1b  96
1ba9 : 30 df e0 82 4f 55 e5 1d  c3
1bb1 : dc 09 79 11 87 a1 0f 67  23
1bb9 : 05 97 b4 9a b2 80 f2 00  05
1bc1 : 7d cf c4 1b 6b b4 37 97  23
1bc9 : 61 49 00 3b b0 02 15 d8  57
1bd1 : 1d 82 1d d8 01 b7 a2 0e  06
1bd9 : 5c 00 ee c0 87 cc f4 07  ca
1be1 : 82 72 55 16 3e 84 d2 32  6c
1be9 : 41 b6 80 1d d8 11 87 6a  52
1bf1 : 1d d8 0b 15 f6 98 78 0a  0a
1bf9 : a4 77 3c 25 e4 46 1e 8d  21
1c01 : 39 bd dd bb 83 9b cd d5  00
1c09 : a3 ca 0d bd 13 8c f1 34  d2
1c11 : d0 c3 6f 45 4f 8b 67 ef  16
1c19 : 24 4d 3c 3b 3f 79 22 62  67
1c21 : 06 be ee cb d0 0f 82 ec  25
1c29 : 0e c1 9d d6 e7 42 5e 44  ed
1c31 : ef c9 9d 09 79 13 be 06  c5
1c39 : 74 25 e4 1c 13 75 68 a4  c4
1c41 : 1b 68 0a 81 ce e6 ea d1  b6
1c49 : 3d 5a 48 41 2f 22 6e ae  09
1c51 : de 7c ee 50 4b c8 06 af  a5
1c59 : c5 52 8e a8 e6 f3 75 68  b5
1c61 : d6 1d ac 09 79 0b 0a cf  ca
1c69 : 4c 50 e5 c0 a9 1d aa c9  30
1c71 : 59 ad 43 37 56 8c 95 9f  b8
1c79 : 4e 87 f9 3e a1 0f 32 05  36
1c81 : 48 e2 26 b4 db ec d8 6b  ba
1c89 : e4 1d 8e 12 f2 16 28 6d  3d
1c91 : e8 83 5d 76 12 f2 21 f3  86
1c99 : 3d 01 e0 d4 26 b8 19 66  83
1ca1 : 2e 4d 95 3a ed ab 27 6b  d2
1ca9 : bf 13 2a 13 ba 09 9b d0  e3
1cb1 : 0e 4c 0a 8c a1 0f 46 9d  e0
1cb9 : 96 31 c1 69 be 87 b5 a4  ce
1cc1 : 65 44 9b da b1 bb 9f d3  aa
1cc9 : 13 4c 9d 9e 98 a1 31 44  21
1cd1 : 37 75 2c f2 d3 68 d1 31  56
1cd9 : ea 21 4f d5 92 ac e7 64  d9
1ce1 : d2 85 cd 52 e0 45 aa e9  ea
1ce9 : b4 64 97 52 23 3e ba 51  b1
1cf1 : 11 27 4d a5 1c e2 74 6e  25
1cf9 : a3 e6 1b 4f 51 94 7d 45  fa
1d01 : 8b c2 69 d6 ab 4c 1c a2  f5
1d09 : 4f 30 d2 c4 5e 09 3d 0b  f7
1d11 : fb b8 f6 cc ef ce 29 31  3c
1d19 : 83 bb 00 29 20 07 76 00  b3
1d21 : 5c d5 1c 22 63 2b 7c 39  a7
1d29 : f5 6f a8 30 84 e7 76 0e  83
1d31 : 51 3b aa 48 72 98 42 66  95
1d39 : dd 5a 70 b9 aa 8c 3d 09  ad
1d41 : c5 53 08 41 c1 bb 5d 2c  a2
1d49 : 3d 54 b6 ed a3 45 4f e0  7f
1d51 : e2 2f e0 cb 6e 31 1a 6e  32
1d59 : 25 a3 0d f8 90 96 7e 25  b4
1d61 : dd 01 ea 81 54 50 cf e5  7c
1d69 : a1 1b c3 58 a1 31 44 37  b7
1d71 : 61 c9 81 0e a5 35 6b 97  ba
1d79 : 6b 07 ba 05 51 58 ab b7  ad
1d81 : 71 73 b4 f8 a6 79 b4 f7  f1
1d89 : 92 34 b4 3d a7 bc 90 70  8e
1d91 : 6e f7 1e fd 1d 70 dd 9d  4a
1d99 : 58 44 27 06 8a b5 06 f3  f4
```

Listing 1. Die zum Brennen benötigte Software für das EX-SMON-KERNEL. Bitte mit dem MSE eingeben

```
1da1 : d8 eb 73 a8 cf c6 1d fc  02
1da9 : 64 cd ef 31 f9 33 a3 bb  55
1db1 : 37 a6 c7 c0 ce 8e 24 ec  11
1db9 : fe 74 9a 60 e0 dd ac 5a  08
1dc1 : 30 67 24 b8 06 de 77 49  8d
1dc9 : 9d 26 99 07 97 eb 37 5d  31
1dd1 : ac 47 90 b5 5a ee 52 2e  be
1dd9 : 80 c8 93 4e 4c 40 e0 dd  72
1de1 : ac 0b a3 b2 f2 97 79 91  47
1de9 : 75 4b 5f 53 63 07 06 ec  a6
1df1 : 0c 89 34 e1 c1 bb da b7  e0
1df9 : 45 20 d7 dd cc 55 aa 25  6c
1e01 : d9 f0 99 bb 55 da e7 01  fe
1e09 : 33 76 a9 eb b6 21 33 76  8d
1e11 : 6d 33 86 a8 d3 ab 56 73  a9
1e19 : 70 fd 37 67 3a 94 3b 50  18
1e21 : af de 1f a6 ed 66 e6 d3  31
1e29 : 2e 04 64 bf 95 ae 76 cd  af
1e31 : b6 74 3f 4d dd 48 ee cd  12
1e39 : 56 cc 50 e1 87 f9 bb a6  ca
1e41 : 40 c8 13 18 3a 2f c4 1b  14
1e49 : 6b b4 37 97 68 58 22 b1  04
1e51 : f3 1e 3c c7 12 a6 62 ce  d9
1e59 : 63 b5 53 31 c3 0d 7a 2b  77
1e61 : b0 c7 38 1a f4 6d 3b 50  8e
1e69 : e5 78 6b b0 c7 77 4d c3  70
1e71 : 36 91 76 42 3a 20 8c 8a  42
1e79 : 35 8a fa e1 52 8d 35 99  88
1e81 : c5 43 bf aa 0f bd 18 30  cd
1e89 : 43 be aa 54 f0 fd f5 46  c4
1e91 : 78 12 f2 0d 2c 12 f2 16  bc
1e99 : 28 6d e8 82 5e 40 7a 78  c5
1ea1 : 2c b2 c3 ef aa 38 f4 3f  d4
1ea9 : ee a8 1c a2 1a f4 69 4f  d4
1eb1 : d2 e9 5b 85 4a 20 93 da  a9
1eb9 : 48 b9 86 f3 3a cf 26 8c  d2
1ec1 : bf ad 2d 51 16 48 cf 2c  08
1ec9 : 7c c9 a3 4b 58 af c5 85  a1
1ed1 : 31 2c 47 93 1d f1 33 4d  25
1ed9 : ce 29 31 82 15 d9 a5 87  9e
1ee1 : 9c 0d b4 03 3c 0e 5c 03  3d
1ee9 : 3c 06 10 11 48 29 20 07  ab
1ef1 : 76 02 ac a9 02 97 c7 06  d1
1ef9 : 33 b3 48 66 9e 86 e1 19  bd
1f01 : ab 2c 69 3d f5 28 23 16  1e
1f09 : 16 28 2a 8d 44 5f ec 1b  99
1f11 : 54 56 6e bc 35 e8 ae fc  13
1f19 : 8d 39 f5 23 cb 46 40 6d  ef
1f21 : 51 54 b0 9c bc 24 b6 28  74
1f29 : 5c d5 2e d5 c5 b4 ca e5  af
1f31 : c5 77 50 84 1b f2 79 93  ad
1f39 : cf 3c b9 63 4b 11 79 9e  61
1f41 : 53 f9 e9 f6 fc b5 ee ba  57
1f49 : cf 1b ba eb 79 93 46 5f  de
1f51 : d6 96 a4 c9 2a 1a cb 50  18
1f59 : fb d1 0d c2 2a 94 3e 7a  0e
1f61 : 25 e1 e9 e0 b4 65 7e 98  af
1f69 : 77 c8 9b cf 19 f4 de f5  c6
1f71 : 37 9e 15 56 ae 74 ab 1e  81
1f79 : c9 da a7 ac 57 46 f3 fa  9c
1f81 : da 72 6e cd ef 75 b1 09  6d
1f89 : 56 1d 8a 29 fd 29 34 a6  7d
1f91 : f1 78 be 79 b5 76 3e f4  8f
1f99 : 5f 11 d7 07 c5 fe 39 34  f9
1fa1 : 3b 8f 17 83 ab fe 3f f7  7a
1fa9 : 43 b8 d5 c2 75 92 ee 69  90
1fb1 : 75 63 29 cc a6 f3 51 75  f6
1fb9 : e9 d4 2b 79 c1 dd e5 f9  9d
1fc1 : fb 87 93 83 b5 4c 3c ed  60
1fc9 : 1a 1f 4d af 17 de 19 ef  69
1fd1 : 7c f8 b7 bc 5d 58 cf 7b  1e
1fd9 : e2 b2 c0 25 2c 27 5d 6d  35
1fe1 : 3c d1 a9 7a d8 89 87 7c  b0
1fe9 : 88 7d e8 85 55 a8 1d 83  31
1ff1 : 0c e6 73 75 68 f7 af 9e  3e
1ff9 : 1a 58 1b 5a b9 dd 86 de  b4
2001 : 88 9a 50 d9 3f 12 c2 06  c1
2009 : d6 b1 a1 3a b1 dd da 3c  55
2011 : 79 a5 67 e7 02 4f 42 92  fc
2019 : 04 dd 5b 3d 10 6d e8 a6  e8
2021 : a1 86 de 88 1b 5a da 3c  36
2029 : 85 53 68 f0 37 97 61 57  f4
2031 : 82 cb c1 4c e6 ca bb dc  00
2039 : 62 46 1d b0 6b d1 56 7d  b5
2041 : 8e ba 85 19 68 c3 d1 b9  10
2049 : f7 c3 c9 a2 9c 0b 9a a5  c0
2051 : c0 8b f1 2e 94 1a 41 f7  28
2059 : a2 09 3d b2 f7 c3 6f 44  09
2061 : 1e 2d 10 dd 54 b3 cb 0a  fc
2069 : 48 01 dd 81 65 8d b5 9b  aa
2071 : 65 24 1b fc 10 43 04 12  9e
2079 : 7a 1e 9e 0b 2f 35 6e f5  4e
2081 : 0f b3 06 77 fc 50 93 d3  23
2089 : 9d fe 14 f4 f7 e6 91 9e  69
2091 : 0f 26 8a 30 f4 21 f7 a2  d9
2099 : 07 60 83 57 e2 37 a7 37  91
20a1 : 56 39 2c 51 f1 cd a6 1f  2f
20a9 : 88 8b 5b 2f 7e cf 4c 50  ec
20b1 : f1 68 87 a2 a9 1e 5c c6  97
20b9 : ec 37 ef c4 14 b8 21 57  0f
20c1 : 82 cb 7c 98 6f 2e d1 cb  a2
20c9 : 0e 55 15 d7 09 29 df 79  0e
20d1 : 2b ad 72 41 af 45 6c e2  34
20d9 : ee 78 cf 58 a1 c3 46 eb  2b
20e1 : de 4b 41 75 fa 92 4c 94  02
20e9 : af dd e9 76 3a e4 bc 5e  4a
20f1 : 12 4e 33 83 a7 35 2e c7  d4
20f9 : de 4b c5 fd 49 38 ce 26  8c
2101 : c3 30 44 96 60 d4 44 78  6f
2109 : 99 9e 21 b1 47 40 bd 6a  b2
2111 : db 3f e1 45 b4 a4 15 4b  08
2119 : 3b 56 98 a8 52 09 85 e8  10
2121 : 88 29 37 8b ce e3 10 4d  64
2129 : 13 5f a0 5e b6 a2 cf 7b  96
2131 : 45 b4 a4 15 4b 3b 56 98  b5
2139 : a8 52 09 85 e8 88 26 96  96
2141 : 8f 37 cf 1b af 30 94 bd  0d
2149 : 11 dc 62 08 e6 c4 2b 4a  38
2151 : 44 e1 2f 57 75 3b c5 a2  ca
2159 : ba a4 14 66 76 ad 26 90  c6
2161 : a4 13 0b d1 10 5b 9f 17  14
2169 : a3 e9 af 30 94 bd 11 dc  28
2171 : 62 08 e6 c4 2b 44 d7 c2  e3
2179 : 5e ae a2 76 05 15 d5 20  36
2181 : a3 33 b5 69 34 85 20 98  79
2189 : 5e 86 82 69 d1 79 f8 3c  3e
2191 : 6e bc c2 52 f4 47 71 88  b9
2199 : 23 9b 10 ad 29 13 84 bd  fc
21a1 : 56 73 bd bd 0d d5 20 a3  1f
21a9 : 33 b5 69 34 85 20 98 5e  10
21b1 : 88 82 dc bc 5e 77 18 82  50
21b9 : 68 9a a5 73 35 41 c8 10  e7
21c1 : 77 51 99 c0 15 06 38 95  ed
21c9 : b6 e9 4e 1c 8d d1 45 38  78
21d1 : 77 e2 ba 28 95 f4 16 bc  40
21d9 : 0c d3 d5 3b 62 09 a9 0b  d7
21e1 : b6 23 9f f6 3b 62 1b b5  8e
21e9 : db 61 a3 cd 3b 1c 25 e4  0a
21f1 : 30 c7 12 f6 18 02 53 1c  7f
21f9 : 41 d1 80 d1 4c 71 07 dd  a5
2201 : 00 ef ea 12 f2 03 bb 00  2c
2209 : 0e c1 0e 03 f1 04 2b b7  37
2211 : d9 b5 b9 41 af 90 76 38  25
2219 : 4b c8 35 5a f7 0e c7 f4  5a
2221 : 34 5e e0 f1 ff 36 8e 04  ef
2229 : 3f ff 9b 4c b9 d3 2a c5  47
2231 : c1 15 24 3e 0c f2 b3 ec  4f
2239 : 6a 1b 23 49 19 9e 6c 19  0e
2241 : d7 2b 4e 6d 33 1a 86 c8  9f
2249 : ee a3 0f 46 12 d5 c0 f6  56
2251 : f0 37 97 61 5f 45 65 8c  be
2259 : 8a 40 93 d4 1d 0c 2a  c4
2261 : 79 a9 c9 dc 46 cb e3 4e  ac
2269 : e2 36 5e 6a b4 da 1b 4c  72
2271 : bb 54 8e b5 0f 9e d9 a0  3f
2279 : 7e 4a e7 46 78 89 f9 22  df
2281 : df e8 84 47 8c 8f 95 96  a7
2289 : 3b fc 84 a5 70 4a e7 90  32
2291 : cf 5b 07 e4 ae 35 5b 8b  85
2299 : e1 bb 38 a4 96 74 67 88  b6
22a1 : 9f 92 2d fb 31 11 e1 aa  cd
22a9 : dc 5f 0f 5e 8f 50 28 f3  c8
22b1 : c1 2d 5c 15 f4 48 7a 3a  b2
22b9 : 10 fc 95 c1 d4 cf 20 be  af
22c1 : d8 28 f3 d3 38 02 a0 d5  e6
22c9 : 08 25 ab 83 f7 5b 0e 8d  6d
22d1 : 83 51 3a c9 3e 67 00 57  92
22d9 : 53 25 8c c3 30 59 25 ba  32
22e1 : 99 2c 6a 0c 01 50 6a 84  f2
22e9 : 98 43 b6 0e 7d 70 4a e7  e7
22f1 : 83 53 44 12 d5 c1 0e da  cb
22f9 : 87 00 54 38 02 19 c1 f0  6e
2301 : b6 0e 7d 70 4a e7 83 53  c4
2309 : 45 06 0c 10 96 ae 0f 85  7d
2311 : b4 ce 00 a7 27 27 27 66  36
2319 : 4a a5 68 a8 30 04 3f 8e  a2
2321 : 67 00 53 93 93 b3 25 75  26
2329 : 92 22 3c 8d 3b 32 5c 92  69
2331 : 92 59 35 84 1b 7a 29 a8  c9
2339 : ce 5e 1c 52 4b 1a ad c1  47
2341 : e3 6e 67 8e b7 ec c4 47  8b
2349 : 81 cd cf 06 a6 89 0f 0b  6e
2351 : 42 12 d5 cc 33 82 0a dc  d4
2359 : f2 1f 8b 43 33 80 2b a9  5d
2361 : 92 c6 a0 c0 12 61 2d 5c  30
2369 : c2 b4 4e 2d 57 d2 4c e0  be
2371 : 0a 72 76 64 ad d4 ce 09  ad
2379 : a2 95 a2 a0 c0 12 1f 8b  d3
2381 : 42 12 d5 cc 2b 44 15 b9  78
2389 : e4 3f 16 84 25 ab a6 70  4e
2391 : 05 39 3b 32 55 2b 44 87  17
2399 : d5 9e 09 6a e7 f3 d0 58  5f
23a1 : 60 09 ba 8c c4 d1 4a e1  6e
23a9 : 21 f8 b4 21 2d 5c c1 98  86
23b1 : 94 c0 12 1f 8b 42 ad e3  57
23b9 : 97 93 8e 65 96 28 6d e8  9c
23c1 : 97 c3 0d bd 11 25 9a 11  fc
23c9 : e0 73 eb 82 57 3c 12 d5  f9
23d1 : cc 2b 45 24 0a 56 88 2b  d5
23d9 : 73 c1 fb ad 90 f4 2d 95  72
23e1 : ac 63 fe db 22 94 61 e8  98
23e9 : d8 56 8b d3 db b6 fd 9a  ea
23f1 : e6 c3 9e 06 75 c8 78 5a  55
23f9 : 10 d9 3f 11 25 9c bc 3a  86
2401 : 74 25 2b 82 57 3c 82 fb  7c
2409 : 63 bd 41 9e 77 f9 5b bd  9f
2411 : b4 df 0d 20 e7 d7 02 39  b4
2419 : e3 4b 07 2e 01 9e 21 c7  43
2421 : ae 58 f0 db d1 2f a9 0d  0a
2429 : bd 13 a5 e6 ab 70 7a 33  44
2431 : ed 37 fa 21 2f 68 47 8d  0b
2439 : 32 4a 49 64 c3 9f 5c 67  e9
2441 : 8e ba de 82 f3 4b 64 94  89
2449 : 39 70 1a 07 2e 00 6f f0  a4
2451 : 50 49 e9 df c9 6e f6 d3  50
2459 : 7c 9a c2 0d bd 15 46 50  b3
2461 : 97 c3 0d bd 16 51 49 2c  3e
2469 : d0 8f 13 09 a9 13 46 b4  9d
2471 : 8e 92 08 23 29 c8 13 34  3c
2479 : 90 ff cb c8 48 c1 5c 03  1f
2481 : 49 0e cd e2 ed b0 db b4  de
2489 : 68 1a 9a 24 3f 16 85 76  d1
2491 : 94 7a 69 9c 01 55 6a 84  be
2499 : 3f 71 a3 43 69 97 79 23  62
24a1 : 9e 0d 4d 12 1c dd 09 9e  6e
24a9 : 13 56 69 62 79 e7 20 4f  84
24b1 : 14 2f ef 2c b1 91 48 12  cb
24b9 : 7b 33 80 29 c9 c9 d9 92  8b
24c1 : ba c9 11 1e 46 9d 99 29  72
24c9 : 25 90 e9 5b 2e ee a2 92  26
24d1 : 12 d5 c7 36 53 90 25 8a  ea
24d9 : fb 36 1a f9 01 0a ed 5f  8c
24e1 : 79 6b c0 cd 3d 55 b8 5c  14
24e9 : d5 2e 04 68 97 5b 8e f4  5c
24f1 : 3b eb c0 55 15 30 c4 0a  f7
24f9 : a2 a7 1c 47 33 1f a6 bb  9d
2501 : 3e 36 a1 df 5e 02 a8 a9  ab
2509 : c7 11 c0 39 45 db 11 b5  93
2511 : 0e fa f0 15 45 4e 38 8e  40
2519 : ab 1f 9c bb 58 33 81 54  c0
2521 : 54 c3 10 2a 8a 9c 71 07  02
2529 : 7d 78 0a a2 a7 28 47 5f  51
2531 : 28 02 35 a3 34 90 49 ea  df
2539 : 08 23 dd fc b4 ba 7f 40  89
2541 : 32 35 3a db 70 a5 96 d5  52
2549 : 95 ad cb cd 22 1a 9a 5d  79
2551 : 63 c0 a5 db 05 ce b7 59  51
2559 : bb 0d 8f 2f 6c d4 a1 73  3f
2561 : 54 bb d3 47 bf 68 d1 0f  15
2569 : c2 b9 80 0a 3f 42 9e 3f  68
2571 : 56 9e 3b a6 19 82 ef 30  80
2579 : c4 3f 75 83 51 77 9c 71  d1
2581 : 0f 64 71 54 e4 09 09 6a  39
2589 : e6 0c c5 de 71 c4 3f 39  6f
2591 : 1a 16 1a 21 21 ed de 30  be
2599 : e3 88 9e 2b 0c 31 13 56  11
25a1 : b1 64 fc 5e f3 94 24 7c  fd
25a9 : 29 4e d8 99 f3 d8 5e 89  f5
25b1 : c2 4b 92 1f 1c da 65 cc  69
25b9 : d1 a7 43 d8 bc 41 c2 2b  81
25c1 : bc 91 e8 8d 2c b9 68 58  0c
25c9 : 0b b6 ac 9d ae a3 64 92  cd
25d1 : 1f 75 72 6b 14 36 f4 5b  32
25d9 : 4f 14 b0 f9 9e 80 f2 ce  f5
25e1 : 4f 69 e2 95 6f d7 66 ed  7b
25e9 : b6 d2 90 55 2c ed 50 a4  94
25f1 : 13 0b d1 10 4c 2b 44 bf  af
25f9 : 81 a2 14 cb 31 30 ce 0a  2e
2601 : 90 3c 52 b7 e6 f5 d4 ce  4a
2609 : 0a 81 98 b6 7a 21 34 52  f7
2611 : b4 46 12 97 a2 3b 8c 41  19
2619 : 1c d8 85 4c ba af 27 97  81
2621 : 1f 8f 29 b2 ee 74 3f d7  e8
2629 : bb 51 79 89 6d b1 b6 dc  15
2631 : db 7c 96 df c3 3d ec e7  95
2639 : 85 9e 75 9e d3 e7 bf ef  9a
2641 : 3d 91 9e f2 b5 1c 87 5e  64
2649 : 27 9d ba ed 9e 7d 58 af  42
2651 : af 85 56 53 ea c5 7c fe  90
2659 : 48 b4 17 a2 72 35 4b 41  96
2661 : 5a 24 cd 26 bc de 0a f3  d8
2669 : 76 17 9b b2 bd 85 49 e8  a7
2671 : 6f 36 0d 09 d9 68 4e c3  01
2679 : 42 78 2d 09 26 74 ba 93  7c
2681 : 62 42 7f c3 08 76 c6 79  9f
2689 : 51 94 22 3c 34 b6 51 49  05
2691 : 2c 9b f9 28 2f c0 ab ce  53
2699 : de 28 d5 8b 06 ac 5a 55  4c
26a1 : af 60 d6 c1 a5 5f e7 37  d1
26a9 : 95 f0 58 b6 f9 f3 c8 b6  73
26b1 : 27 24 33 9e 39 bf ba ea  dd
26b9 : 96 49 07 52 5c 38 5e 39  73
26c1 : ab 1d 64 f4 51 7e 2a 8c  7d
26c9 : a1 2f 56 7f 83 93 d9 fe  c2
26d1 : fb 8f ef ed e4 c6 67 96  9d
26d9 : 6e 49 3c 60 c8 e2 07 67  95
26e1 : c0 3b 98 b7 80 1b 1a 33  0c
```

```
26e9 : 8a 4f b1 a0 59 cb a6 4f  c8
26f1 : f1 31 fb c2 37 68 ef a0  8a
26f9 : 29 25 c3 f6 51 9d f4 2c  b3
2701 : 54 29 25 c9 fe 26 3f 78  7b
2709 : 7c 5d a9 3f f0 4f db 65  4a
2711 : fe 21 e8 06 7c af ef 27  ee
2719 : f8 b2 fd e1 49 2e 6f f1  cd
2721 : 65 fb dd 64 cf b2 14 32  cf
2729 : 7f 89 8f de 1f b7 88 77  ed
2731 : fb 58 b4 df b4 6f 53 d0  37
2739 : 7f ac 4e fc ad 29 24 e5  c2
2741 : 8a 1b 2f 1e 3f b2 30 07  41
2749 : e2 6c e7 0e 63 94 b3 b9  3a
2751 : 3c 32 77 d0 e4 f0 c9 b7  0b
2759 : 59 bc 9e 19 33 50 e4 f0  06
2761 : c9 d6 43 c3 a3 3b e5 58  3b
2769 : a0 af 68 e5 16 a3 0a 49  f1
2771 : 73 bf aa d4 61 fb 78 8f  00
2779 : ba c6 ef 89 d6 73 df 3e  c8
2781 : cd 1b 6e 4c d3 9d b9 c5  9e
2789 : 27 a2 6f ee 4f 18 eb e2  e6
2791 : 7f 70 f2 38 8a fc ab 15  75
2799 : 0e a7 69 72 23 ac 98 a8  6f
27a1 : 73 ec a1 9a 7d 34 33 d5  38
27a9 : e4 21 ec d9 06 bd b4 59  68
27b1 : c4 c3 e6 6d 17 60 98 30  f5
27b9 : 7b 16 c1 30 a6 6d 17 36  54
27c1 : 98 3d 6d 14 69 84 26 01  2b
27c9 : 6b c3 3b 9c bc e9 65 ec  03
27d1 : 33 89 2f 61 98 94 cb 54  c7
27d9 : b8 15 a9 52 6b ae 0a dd  e1
27e1 : 4c c4 a8 1a a5 b3 ad 49  3e
27e9 : a2 99 c4 93 0e a7 39 f7  ee
27f1 : 72 cb a0 76 c8 39 ad a2  92
27f9 : ee 13 38 52 87 14 94 e0  f6
2801 : e3 3c fb 65 65 28 6a 70  d0
2809 : f0 9b 45 fa 0d 4e 8f 92  1e
2811 : ae a0 38 6d 9c 33 1f 92  d0
2819 : 6e fd 71 e8 d6 1d b7 5b  eb
2821 : a6 87 23 e1 4c 38 6d 9c  05
2829 : 78 08 cf ab e2 21 ec d9  ad
2831 : 1b ee 5c 78 eb 50 85 97  70
2839 : 04 ef e0 b7 4c 39 1f 0a  83
2841 : 1c f3 05 03 e0 f7 4c 87  07
2849 : fc 7c 1b ad 80 6d 5b 45  6b
2851 : c0 4c 37 5b 03 52 87 ae  af
2859 : d9 97 ae 85 15 19 c5 27  da
2861 : 0c 21 e0 ac 20 97 6c 13  62
2869 : db 00 fa 36 01 ce 7a 05  44
2871 : d8 2c 57 34 a1 2e d8 39  1d
2879 : cf 40 bb 08 58 af a9 50  a2
2881 : 97 6c 15 bb 04 3d 4e 18  9f
2889 : 7f 12 8c f6 8d 13 0a e9  81
2891 : 90 fe 25 1d aa 0d fd 53  3f
2899 : 8a 13 ad 83 45 b0 0f 53  45
28a1 : 86 e6 96 12 ae d6 28 4b  5b
28a9 : b6 09 ed 80 17 60 e7 4a  18
28b1 : b1 4d 4b 46 93 21 54 84  41
28b9 : bb 60 9e d8 01 76 10 b1  ce
28c1 : 5f 69 4d 4a 12 ed 82 7b  03
28c9 : 60 15 bb 00 f5 18 07 4f  7e
28d1 : fb 3e c3 b2 e5 21 3a d8  34
28d9 : ea 35 b6 01 fb 78 81 99  e8
28e1 : e3 9d 62 b6 c1 f7 fe 05  e4
28e9 : 67 68 33 3c 70 e1 78 e7  a0
28f1 : 14 9f 60 64 71 0e 82 b6  78
28f9 : c0 2d f8 86 f4 57 b3 6a  0c
2901 : 14 bc 43 7a 31 e4 8d 8f  23
2909 : 4a d8 98 e1 d9 f0 0e 82  84
2911 : b6 c0 3f 6f 11 a3 34 83  eb
2919 : b3 e0 1d 9d 5b 60 86 9e  47
2921 : 8c 23 76 8e 29 31 c3 f7  c9
2929 : 6c 15 db 2c 50 bf a3 3b  a4
2931 : e8 58 a0 af 68 e2 93 d1  f3
2939 : 36 38 58 6c 0e db 2c 54  48
2941 : 33 9d a5 e8 5d 6c 3a 34  54
2949 : b9 cf 69 5f 8e 97 39 ed  96
2951 : 2b bc 7e ec b3 f3 12 be  b8
2959 : 57 73 ce 73 49 59 6d a3  e8
2961 : f9 36 8d 3f a0 be 4e 57  a8
2969 : fa fc 7c e7 26 fd 9c 6a  97
2971 : 4f 8e 97 39 ed 2a 6f 45  8d
2979 : 7b 34 83 b3 e0 1b fb af  9c
2981 : 5d d1 65 58 77 8e b5 98  1f
2989 : 67 3b 47 7b eb 78 61 a7  26
2991 : a3 68 bb a2 ca b0 ea 35  f4
2999 : b6 01 6f c4 68 cd 24 3b  40
29a1 : 3e 02 78 1f c3 b1 f1 ce  11
29a9 : e3 7f 1e 45 db ec bf bb  18
29b1 : e3 cf 45 67 f9 f3 3b 7f  e5
29b9 : b6 37 e7 f9 f2 3d cf c3  24
29c1 : fe ae 36 fe 32 1f f5 c3  ff
29c9 : 83 dd ef ff eb 35 1e a7  67
29d1 : ff bf fb fa e2 6d 35 7f  7b
29d9 : fc 87 0a d3 e2 de 4d 4d  8b
29e1 : 4d 5e f4 5f bb 28 70 9b  7c
29e9 : 1c 52 5e 87 7e fb 3e ad  d3
29f1 : f6 b8 a9 42 8e 92 1a bb  53
29f9 : 24 3f fd 64 19 0f 43 c0  61
2a01 : f0 c3 97 76 ad b7 48 6f  20
2a09 : 65 7b f1 9f 2b f7 30 4e  6c
2a11 : 0a ed 5c 5a 75 c0 8e 76  f9
2a19 : a3 2a 3d 46 50 d3 be d6  36
2a21 : fc 57 39 f3 90 26 a5 46  f3
2a29 : 69 1c bd 6f 05 4d 12 5b  b7
2a31 : 3c 74 8e b4 5b b0 49 f9  36
2a39 : 89 e3 a6 96 93 22 53 0e  e4
2a41 : d5 d4 4e ab e1 ee 3e d8  ca
2a49 : cc f4 67 24 4c 83 8a 46  85
2a51 : 84 bd 23 a5 45 41 5c 2a  55
2a59 : 0c f0 95 dd 39 d5 54 07  a0
2a61 : ea 6c e7 65 31 29 50 1b  fc
2a69 : 46 ce 7d 99 8a 87 3e ec  61
2a71 : c7 eb 58 ba 0b 4d 94 ae  66
2a79 : c8 67 ca ca 26 6d 95 94  4e
2a81 : a0 37 b2 38 4b 58 87 23  4c
2a89 : e1 7f 71 05 0c 47 df 2f  00
2a91 : 87 f1 d9 7e a7 ac f3 cf  a6
2a99 : 7e 07 33 2a e0 4e f9 8f  34
2aa1 : 6e 8a 94 66 76 a9 a1 8b  99
2aa9 : 02 0c 19 9c 01 74 b5 d3  bd
2ab1 : 62 9b 17 8c 88 59 23 d7  c8
2ab9 : 97 a0 c0 13 fa c8 27 36  32
2ac1 : 21 60 c1 d7 4d 89 fc fc  8d
2ac9 : 16 94 81 30 f7 ae dd 79  ef
2ad1 : 8b 24 cc 85 83 05 83 e3  88
2ad9 : dc 55 a3 3c 43 4d 8a de  57
2ae1 : 99 0d 8d ff 61 78 2e 6d  d2
2ae9 : 34 d1 90 a1 26 70 96 bf  1e
2af1 : 0a 2a e0 87 82 85 2e 5a  fb
2af9 : c9 59 df b6 da 3b 4a a5  39
2b01 : 9d ae cf 6a 54 c8 71 c2  0d
2b09 : 27 c2 9b 07 27 da 3f 12  c4
2b11 : 69 87 f1 2a e9 44 4f 68  ce
2b19 : d1 3d 6e eb cb c4 dd 79  ef
2b21 : 5a 29 c6 be 36 64 f4 f6  e1
2b29 : f8 ff 64 53 25 c9 2a a5  39
2b31 : 5a c1 35 69 16 dd 27 16  ff
2b39 : b1 ce 6e 63 18 35 d6 2f  3e
2b41 : b3 76 58 c9 82 d6 27 15  24
2b49 : 4b 43 c9 f3 1c bd 4f d0  b5
2b51 : 79 d2 14 64 79 d4 ea 7f  ae
2b59 : 43 ce 2a 8e 4f 39 da 9f  c9
2b61 : 29 ea 54 9e 76 61 3b c2  cd
2b69 : 70 98 54 9f 25 7d 8a 71  7a
2b71 : 66 bd bf 35 d6 4d 23 21  f3
2b79 : 43 9c a3 f2 86 fc 47 36  2b
2b81 : dc 79 aa ba 06 d9 51 99  c3
2b89 : e8 56 2c c6 34 c6 6d f6  9d
2b91 : 6c d8 dd 9f 36 3b 77 59  a2
2b99 : 34 84 c6 8c ce ec 11 ca  81
2ba1 : 3b bd 9c ec f9 b1 e7 9d  87
2ba9 : 64 d2 4e 33 bf 27 84 84  c1
2bb1 : dc 17 ab 91 49 08 92 a9  a8
2bb9 : c1 05 ac 50 f4 6c 4e 48  ae
2bc1 : 9d 23 8a 4d a2 bc b9 de  f1
2bc9 : 25 55 d3 6f 9a 30 ed 53  05
2bd1 : e0 7f c3 d5 2c ef 5c fb  c8
2bd9 : af f8 7c e3 da 61 e5 d3  98
2be1 : 53 68 10 a0 7e 20 d5 71  a4
2be9 : 8d 4c 4e 12 9f d4 f4 6d  c1
2bf1 : 7a e6 f9 91 cc 37 b1 33  43
2bf9 : f2 1e b1 66 3e 60 bd 82  16
2c01 : 1d 9b 15 df 18 d9 ab 25  76
2c09 : d7 73 4d 3b 2b 55 75 fc  82
2c11 : 0f 8c 3b 54 7e 65 cd 47  99
2c19 : 0e cb c2 98 1c 9e f9 70  50
2c21 : 0d 2a 91 da c3 98 86 b1  81
2c29 : 76 7f d4 f9 8c 69 8e a1  65
2c31 : 97 47 f3 9f 66 c2 d2 c5  b0
2c39 : ea e1 e3 9a 31 31 b8 b0  41
2c41 : d9 53 c5 d9 68 17 c2 5e  77
2c49 : 61 a3 e9 59 63 49 ef 89  75
2c51 : e4 38 dd cf 79 13 7e 43  73
2c59 : 33 e4 af 99 3b 7d d4 1b  46
2c61 : bf 18 f9 2a b9 5c 60 98  21
2c69 : 95 e6 43 1f 92 71 48 33  62
2c71 : 9a be da 51 a4 8c f3 ee  a7
2c79 : 95 1a 48 a4 1a ae 31 cd  39
2c81 : 49 e4 34 9a 69 8a 6f 21  88
2c89 : 77 ce 33 ce 6c e9 35 d8  ab
2c91 : 63 6e 6e a5 28 89 d4 3b  14
2c99 : 0d 39 2a b7 df 12 27 c9  83
2ca1 : 5f 3e e6 a2 7d 37 36 65  62
2ca9 : c9 e1 7b 04 3f b4 ab 4c  a3
2cb1 : 53 73 51 3e 5b a8 d2 69  f3
2cb9 : a3 ac fa 61 d8 9c 95 c6  f3
2cc1 : 37 85 c5 c6 8b bb 5d ad  6c
2cc9 : 47 ea 61 fd 4f 45 24 3f  cc
2cd1 : 6e 53 ec dd 1e e9 95 60  28
2cd9 : 41 9c a6 cd be cd ea bc  4b
2ce1 : de 4f 9b 39 c1 f3 64 af  21
2ce9 : b7 1e 6c 63 78 4e b2 69  ce
2cf1 : 09 e1 70 7c d8 e9 46 1d  c7
2cf9 : 12 08 e5 53 66 c3 f3 ce  e5
2d01 : 1c 52 63 26 fb 36 39 e7  0a
2d09 : 38 cd 1a 0e ea 33 9c d8  dd
2d11 : 6b ac 6c bc 29 8c 64 d7  bd
2d19 : 93 c8 84 bf 40 bc e2 90  c0
2d21 : 76 58 c6 f2 19 a4 87 53  4f
2d29 : 8c b9 a8 d2 42 d2 71 70  78
2d31 : 32 66 b9 36 4e dc 7e 31  f4
2d39 : 9e e3 76 13 12 a8 7a 52  be
2d41 : af b9 a8 a4 86 f6 55 75  ec
2d49 : c0 fa 1a 01 cf 00 b8 c8  9f
2d51 : 7d bc 60 dc ce 54 cc 63  69
2d59 : 04 dc 18 72 7b ea 9b 4d  38
2d61 : f4 39 37 e2 a9 b4 7e 28  a6
2d69 : 7c cf 08 79 d3 82 d3 7c  98
2d71 : c1 c9 f6 8f c4 cf 58 fb  ea
2d79 : a8 8a 48 6f 65 57 63 8c  1e
2d81 : f2 7c 4b af 46 90 7a 36  b9
2d89 : 2b bc 41 f9 30 c3 b1 39  fc
2d91 : 31 8d 53 cc f0 99 49 53  9f
2d99 : 66 c3 f3 ce 2c 59 8c 64  40
2da1 : 02 02 ed 10 fc 47 5a 0c  ad
2da9 : 9a 4c 59 5c df 17 c2 65  d8
2db1 : 54 8c 1a 6c de 55 23 27  d3
2db9 : 64 c6 1e 8c 3f 3c e2 01  fd
2dc1 : 00 3d c9 c4 31 2a 9c 6f  20
2dc9 : 17 6f 06 79 df fa b8 19  33
2dd1 : 24 85 a4 aa e6 a3 49 0b  fd
2dd9 : 49 c7 dd 41 ee d7 5c 0f  e3
2de1 : d9 b2 f0 a4 f7 cb 80 6e  a1
2de9 : f9 98 c3 ac da eb 51 9b  3e
2df1 : 0e b6 55 97 84 13 70 6a  1a
2df9 : 64 f7 d0 ed 37 d5 32 6f  14
2e01 : c5 0e d1 f8 aa 79 98 c3  c1
2e09 : ce 9c 16 9b e6 0e 4f b4  a4
2e11 : 7e 29 5e 6a 01 01 f7 35  6b
2e19 : 19 f9 2a b4 37 b2 ab b4  71
2e21 : 43 d5 3e cd a3 ec d3 66  56
2e29 : cd 8d 51 95 48 c9 d9 31  60
2e31 : 99 f3 c3 5d 62 3e aa ec  fd
2e39 : 71 bb 38 a4 f8 97 78 83  60
2e41 : c7 d0 23 24 1b 99 ca 99  9b
2e49 : 8c 64 02 00 0b 8c 80 40  20
2e51 : 5d 7a 39 e0 c4 b1 40 20  71
2e59 : 49 5f 07 b9 38 86 25 54  40
2e61 : b2 e7 d9 b9 ca 3d d0 d2  34
2e69 : 08 33 9c 1f 35 9d db 9c  ff
2e71 : da 1a eb 15 4a e7 9a e7  14
2e79 : e2 73 d4 73 8c 1f 9e 71  d7
2e81 : 00 80 a9 e7 17 8b 8c 8d  44
2e89 : 63 ac 9a 48 c3 bf 4e 9b  9d
2e91 : 36 9a a7 9d de f1 71 91  18
2e99 : ac 75 93 49 18 77 e9 be  70
2ea1 : cd d1 ee 91 3e 57 36 9a  f1
2ea9 : b9 c6 3e d4 e2 2e e3 33  85
2eb1 : d1 a7 40 e4 f4 d9 4f ea  34
2eb9 : 3f cf ea e4 76 fd 4e 8e  e5
2ec1 : e3 f4 ef 74 1b 4e 97 d2  d1
2ec9 : 6e 02 f3 30 5e 69 95 e6  91
2ed1 : f5 57 aa da 9b 37 4c b9  90
2ed9 : bf 5d 6a ed 99 6c 18 0a  11
2ee1 : c2 3b 73 8f 50 b6 1c 01  3d
2ee9 : 58 52 43 46 70 34 d2 18  28
2ef1 : 0a c1 75 b0 eb 21 8e ba  c7
2ef9 : 56 98 74 46 f6 a4 af 89  e8
2f01 : ed 4e 02 39 54 6e a2 f9  f4
2f09 : 33 b0 2a 89 c0 e0 f9 ae  a8
2f11 : 37 9d 9f 69 99 f2 57 db  72
2f19 : 8f 37 40 10 f2 a3 f3 81  75
2f21 : b6 53 08 05 92 be 24 09  e5
2f29 : 5c db ec db 42 7a ad 09  ea
2f31 : 32 d0 8c 1a 13 80 9b ec  af
2f39 : dc e7 07 cd 67 6d 40 63  2e
2f41 : e8 dd 23 5a cb 30 20 ce  88
2f49 : 4c 98 f8 d4 d9 b4 3d a9  45
2f51 : 53 70 17 99 82 f3 4c af  2e
2f59 : 37 aa bd 5d d4 60 47 2a  c2
2f61 : a8 c7 b2 ad 47 32 77 3b  a9
2f69 : 54 4e 07 07 9a 10 f2 a3  c4
2f71 : e5 03 6c a5 7c 9a a3 1e  0f
2f79 : cb 39 c1 f6 9c 6f 3b 3e  df
2f81 : 6b 33 e4 af b7 1e d4 61  37
2f89 : d5 4c 98 f0 f6 a5 50 cf  46
2f91 : e5 38 49 ad 49 e4 60 14  00
2f99 : 79 4b 77 b2 7c 0b 4d f4  2b
2fa1 : 9c cd a3 f1 54 b3 cb 4c  f5
2fa9 : 26 e0 9b 80 49 93 30 70  09
2fb1 : 94 97 34 9e aa 73 a8 d5  86
2fb9 : 13 7d c1 f3 5c 24 d6 a4  05
2fc1 : f2 6b d3 5b bd 04 72 82  94
2fc9 : 6e 09 c1 69 be 0f 69 ec  3d
2fd1 : 1a 96 79 69 e2 9c 79 3b  31
2fd9 : c0 9b 83 a0 b2 c6 b4 df  50
2fe1 : 42 b4 7e 24 cd be 23 f1  e5
2fe9 : 1b 5e e1 29 2e 69 2e d3  5f
2ff1 : 08 2d 8d c2 1b 8a 6e f0  ed
2ff9 : 7d 53 ae 09 de 59 63 4b  c9
3001 : 7c d3 3f 21 f2 7b e8 ba  7f
3009 : 29 5f 98 9e c0 4d c1 93  80
```

Listing 1. Die zum Brennen benötigte Software für das EX-SMON-KERNEL. Bitte mit dem MSE eingeben

```
3011 : 7e 25 3e 43 e6 5c 9e 6f  c4
3019 : 62 96 f8 35 0f fc e8 21  6a
3021 : f2 41 60 d6 c1 a6 2b 49  37
3029 : a6 95 67 15 45 0f 5a ae  aa
3031 : 63 dc 50 b6 13 22 b0 a3  3a
3039 : 6c 8f e6 c5 3b a2 4c b3  40
3041 : a4 12 fc eb c5 0b 60 ab  38
3049 : 75 ec 05 61 de 0a c0 7a  d8
3051 : 3d 2c 5d 52 f9 3c b5 d3  46
3059 : f1 73 17 24 8b 76 aa 1e  a2
3061 : a2 87 b3 3e 63 dc 69 0f  5c
3069 : 6b a4 8b 81 18 7f 8e a1  34
3071 : a7 3d 42 d8 0f 84 d1 05  c9
3079 : b3 d6 5e e4 9e 5c d7 33  5e
3081 : 8a 8d 62 d9 9b 3d 99 e7  7f
3089 : 6f 1c 6f 17 6f 11 8e b1  e2
3091 : eb bc 44 70 97 68 91 d1  a0
3099 : 4a 42 3e 3d 3f 8d e7 ae  99
30a1 : b8 45 da e0 1f ba b3 bf  e5
30a9 : 63 66 7b 39 4a 1d 99 fe  37
30b1 : c7 ca 78 e6 f6 7f 29 f6  d6
30b9 : 54 d5 2c f5 16 5e e4 c7  39
30c1 : 17 80 a1 fb 3a 03 e8 db  17
30c9 : 23 5b 09 f5 35 57 f4 3e  f9
30d1 : 2c f5 d7 fc 3e 71 cd 31  96
30d9 : db c5 0b 60 9b 05 0f 5c  3d
30e1 : 90 f0 bb ca 03 d1 a1 61  39
30e9 : 68 7c 7f 11 83 a1 d2 9e  5f
30f1 : a3 a1 a1 fc 24 53 ae 6e  e1
30f9 : c8 f3 a7 a6 a7 7f 8c 82  27
3101 : 65 ce f7 f2 09 7e 85 1c  fc
3109 : 1e 8c 8f 5a 42 55 5d 7a  d6
3111 : ab 86 be 25 49 1c fc 9a  f2
3119 : 2f 77 e8 d2 7e bf cd 90  d6
3121 : f0 5f ea e7 13 eb d3 cb  f0
3129 : 6a 73 fe ce c7 cc a8 d3  13
3131 : 2f fc bc 9f 16 97 4b 1f  8b
3139 : 59 e8 6b 76 db 2b 74 3b  0f
3141 : 9d 0d 5d 5c a7 fb e7 f9  36
3149 : f7 a2 f4 ff f7 97 f9 2b  49
3151 : dd fd 40 7f 75 26 de da  e7
3159 : 97 5b 3f e9 0b e5 b4 e5  29
3161 : 79 1f 84 df b5 8d c5 17  94
3169 : 57 5f cc f2 fd 7a 6e a7  be
3171 : 37 bd f3 7e 2a 4f 5a 1f  18
3179 : 85 9b 70 ba 34 7e f3 dc  00
3181 : f9 bd 8f c4 3e 13 1f 7f  cd
3189 : f9 cb fb c7 6d 7f 67 a4  19
3191 : 2f 4f e7 dd f1 eb dd e1  d7
3199 : ac 7e fb e3 6d ed a9 75  d7
31a1 : b3 fe 90 be 5b 4f d3 d8  00
31a9 : 8b bd 61 33 26 2d 07 83  c1
31b1 : c0 da 78 fd ae 9c ef 4b  e2
31b9 : 86 6f a7 c0 ec f1 3f 93  7b
31c1 : 85 78 31 fe 0d cd 9c d3  08
31c9 : ef 57 fb 58 7b ff ce cb  f8
31d1 : de c5 fc 7f b3 d2 17 a7  3e
31d9 : f3 ee fa 35 ee f7 0b 1f  c2
31e1 : f0 fc 6d b1 58 3a 1e d6  5e
31e9 : b4 6e e4 8f 1c f6 d6 80  d5
31f1 : 6d dc 57 28 e8 68 04 0d  a3
31f9 : ab 71 d6 1d 7c d6 d0 f8  6a
3201 : f2 5b 59 21 b7 3b 79 dc  10
3209 : b4 3d 4b 43 f6 bf a1 0e  a7
3211 : fe b9 fb b6 c4 6c db f4  ca
3219 : 6a 3c 91 7f 9f 2a f6 7d  18
3221 : b7 ea c7 77 c8 4a 7a e1  bb
3229 : d7 6f 75 59 ea 1e df 0b  76
3231 : 55 8f 25 4f fd 5a e0 67  86
3239 : 2b 5b 33 cc 6c db ce 2b  af
3241 : 6d 43 6f 11 b3 6e df 21  be
3249 : b3 29 e6 d3 ed 9b 68 9b  59
3251 : 36 2b 30 8f 57 6e 9f 8d  9d
3259 : 7c c9 68 bb 51 d1 a2 5f  38
3261 : a7 d9 29 4a dc b7 19 c7  08
3269 : f3 c5 9e ce 08 f4 9d 94  88
3271 : 47 a4 ee 7e 7a b1 0d 6e  dc
3279 : 93 11 ad cf 85 ab 6a 5a  0e
3281 : 4b 7e ba d0 d1 75 56 d1  9a
3289 : 97 12 8d bc 7c fd f6 28  08
3291 : f4 60 bf c6 d0 6d 64 aa  dd
3299 : e5 24 80 2c 91 5f 37 ed  03
32a1 : b3 a7 a8 f4 7a 60 c2 8a  bb
32a9 : 5b db 96 e9 b4 26 e3 ae  3e
32b1 : e1 d9 3a 61 bd a1 0d c6  e4
32b9 : 79 81 58 c9 37 50 ac 7a  e0
32c1 : e1 41 42 1b da 10 ec 50  b9
32c9 : f5 c3 7b 42 07 50 c5 db  89
32d1 : 81 28 66 fa 6c e9 c0 ea  4e
32d9 : 16 56 c0 75 0c 5b 60 3a  8b
32e1 : 85 95 b0 50 50 9c dc 86  d1
32e9 : 60 56 3d e1 58 8f 47 4e  bc
32f1 : cb 77 e8 ee 22 cf 82 d4  e4
32f9 : 23 48 37 b4 20 4a 17 ee  33
3301 : e0 b0 2b 17 1f 8e 49 9a  a8
3309 : 85 63 4c 61 f5 0e 76 a5  74
3311 : 65 81 db a5 33 96 0c 1a  2f
3319 : d8 30 6b 60 e0 f1 d3 81  e0
3321 : d4 31 6e f3 e9 9d 7d 28  7a
3329 : 72 48 6d 74 39 f7 4c b4  97
3331 : a4 dc 07 d7 11 81 58 b8  f0
3339 : b5 50 ac 53 26 e0 3e b8  80
3341 : 9d 70 5a 84 3f 94 40 25  21
3349 : 08 1d 43 1e db 3e 9c df  56
3351 : b7 1e 7e a3 d1 f6 c7 3e  9c
3359 : bc 18 51 34 26 e1 d2 7d  b4
3361 : 70 a0 a1 60 56 32 4d d4  6c
3369 : 2b 13 ab e5 0e dd 86 8a  c4
3371 : 20 7f 28 90 95 a7 8c 3a  aa
3379 : e9 a0 25 0a 1b da 1e bb  b5
3381 : 02 b1 71 9d 50 ac 50 2d  72
3389 : b4 13 84 a5 65 81 db a5  ba
3391 : 33 96 01 bd a1 60 d6 c1  03
3399 : 83 5b 07 07 8e b0 0e a1  56
33a1 : 8f 76 cd d1 8c 8c f9 c7  bd
33a9 : a1 e0 c4 05 a8 40 94 27  b9
33b1 : 55 86 a2 f2 54 c3 8c 73  cd
33b9 : 72 19 81 58 f7 85 62 3d  d3
33c1 : 1d 3b 3c bf 46 c9 81 58  ec
33c9 : f7 85 62 3d 1d 3c 7c ba  de
33d1 : 63 0d e8 2d 43 36 ca dd  67
33d9 : 20 9e a1 35 96 7d d2 6c  d1
33e1 : 0a c5 c6 75 42 b1 4d 81  18
33e9 : 58 c9 33 50 ac 53 60 56  90
33f1 : 2e 3f 1a 85 62 9b 02 b1  65
33f9 : 92 05 0a c5 36 05 63 de  20
3401 : 15 88 f4 74 e9 c2 57 89  4b
3409 : 18 75 d3 49 9b b2 8f 99  bb
3411 : 1f 77 2d fd 95 2b 21 46  bb
3419 : 84 f2 74 93 b4 dd 94 67  01
3421 : a1 38 76 4e 0a 67 04 a7  81
3429 : b6 7a 37 11 f5 1e 35 c1  b5
3431 : a7 46 28 d4 dd 96 1e 91  ce
3439 : ec fc 3a 6e c8 ff 1c af  5c
3441 : 64 68 9a 75 18 83 83 e3  a2
3449 : 65 5b ca ca f6 55 6c 3f  b2
3451 : 71 87 b6 7e 95 95 64 a0  dc
3459 : 98 b3 f1 2d f9 07 14 93  3c
3461 : a4 e0 cc e8 0f 73 d9 e0  7b
3469 : 5d 1f c2 f5 0f 46 f6 4e  61
3471 : 88 9f da 86 e8 f4 ce ab  19
3479 : 94 39 9b 5d e9 ea 6f 4f  8f
3481 : 38 4a 35 63 ac dc 9f 25  13
3489 : 7c a6 e4 f5 1a b1 4d 59  67
3491 : 23 82 53 d2 3d 11 46 b7  09
3499 : 3e 34 64 df 1c 3f 76 9b  d3
34a1 : 15 1f ae 3b 53 7c 0f 3e  2b
34a9 : 12 7c 32 df 07 a3 [illegible] 9d  1b
34b1 : 36 29 be 04 7f 6b 0f 48  cc
34b9 : fa 73 85 d6 c5 1e 74 e6  96
34c1 : 4a 35 62 d7 72 7a 9b 93  ca
34c9 : c8 71 c7 a1 e7 e7 4d f0  45
34d1 : 0e 0a 18 cf 78 1f 49 f0  6c
34d9 : f8 5f a8 f4 79 fb 3f a4  07
34e1 : fa 93 4d 4c e8 0f a3 92  3c
34e9 : 45 de 96 70 7a 3b 4b 53  a6
34f1 : 82 53 db 3d 1e cc 79 33  50
34f9 : 7c 0f 3e 14 52 f1 8e 94  27
3501 : b7 7c 70 a5 0e 77 28 75  ef
3509 : f4 a1 c7 b5 de 9e a6 f4  de
3511 : f4 3f fb 3a 69 a1 c0 46  1e
3519 : a8 9d 26 1e d9 fa 56 4e  49
3521 : 0e 3e 50 4c 59 f8 83 06  63
3529 : 1b 93 dc 1e 3a 29 41 31  5d
3531 : f1 ad 98 09 8c 91 ea 09  53
3539 : 8c a0 98 f8 d5 91 e8 8e  05
3541 : ab 53 7a 7b a4 c2 91 1e  87
3549 : 07 a8 7c 3e 15 d1 e8 dd  ca
3551 : ec fd 43 ea 4d ea 5a 51  a2
3559 : 30 dc 9f a5 8b c8 39 9a  ad
3561 : 95 ac 58 6d 0f 51 db 16  27
3569 : 1d 23 d4 9f 62 77 44 3c  ac
3571 : fc e8 4f 67 4e d1 29 5a  6f
3579 : c5 32 6d 18 7b 67 e9 59  03
3581 : 52 38 6e 8c d9 c7 ca 09  36
3589 : 8b 36 08 e1 b0 dc 9f a5  29
3591 : 8b a1 60 26 2e 2b 94 13  7e
3599 : 16 02 62 e1 da 34 47 32  56
35a1 : 74 87 9f 9d 61 b9 3d c6  db
35a9 : 3a 2f c4 e9 36 1b d3 f8  67
35b1 : 1f a8 fe 17 d2 7a 38 0e  c5
35b9 : 2d 54 de 9f 27 6b 43 fa  8d
35c1 : 8f 4e 49 ba 9b d3 ce 2a  09
35c9 : 6b d8 6e 4f 47 44 c3 72  b0
35d1 : 7b 84 87 44 73 35 1a b1  a5
35d9 : 38 aa 6e 4f 68 9b 5a 5e  75
35e1 : 5e b4 1e 5c dc eb 6e 43  1a
35e9 : a6 d7 21 e6 92 e7 4e be  bf
35f1 : 97 6d e6 2d 2d ad b9 5b  7c
35f9 : d2 43 5b a4 c4 53 d2 b9  7e
3601 : fc bb 65 1a e4 69 39 2f  54
3609 : 15 60 b4 18 30 4d 14 9b  73
3611 : 23 49 92 50 de 0a 87 a4  2d
3619 : 6e c2 89 24 1d 4c 1d 71  5b
3621 : e9 35 43 5b a4 c4 a5 db  a0
3629 : 7c 4d 72 34 9c 9a 5d b7  f2
3631 : 98 b3 5e 4b cb 9a 59 6d  76
3639 : 8d 2e db e2 38 44 ef 10  b6
3641 : c5 95 6b ad f4 14 9b ba  35
3649 : 66 d6 ce 75 16 ee 37 81  35
3651 : af 6d 0e 91 9b f1 3a 4c  37
3659 : db 41 b7 35 32 39 ce ba  07
3661 : 1d 9b 15 c7 8d d4 c6 f4  0f
3669 : 1c a3 65 bd 74 99 b6 83  5e
3671 : c0 36 31 23 1b 36 45 c8  07
3679 : 7d 52 ab 92 1c 91 06 6f  a2
3681 : 08 90 d0 fa a5 62 d4 69  f8
3689 : 59 55 d1 90 73 e1 e8 b6  6b
3691 : 1f 7a f8 de 3f 7d 02 ce  8d
3699 : 75 16 6e d7 7a 43 d5 05  d3
36a1 : 0c 08 63 80 e0 f1 e7 28  28
36a9 : c8 3c 04 3f 73 d7 3f 52  10
36b1 : 70 60 b8 de 4a 18 c8 7d  df
36b9 : 35 c6 64 8b b3 9a ba 4d  71
36c1 : 0b da 10 db 40 8b 6a b8  34
36c9 : a3 c1 61 bd 3d c2 63 ca  6a
36d1 : 46 85 79 ba 4b 92 1b b6  b3
36d9 : 88 6a b8 cc 9d b4 41 8c  fc
36e1 : 4d 2b 48 be 11 3d 48 4b  60
36e9 : f4 0b d9 71 ea f8 e5 2e  72
36f1 : a0 be 59 3c 83 74 97 9c  42
36f9 : 91 3a 46 f2 10 e7 e0 ba  d0
3701 : 69 8a d2 68 de a3 23 ce  a6
3709 : 2a 61 e4 a2 86 7a 00 67  7c
3711 : 1f 89 14 90 15 a8 76 df  3c
3719 : 88 d2 e8 ea 97 24 8e 3c  ef
3721 : 5b 85 c7 8e b5 0d 62 21  92
3729 : af 6c d0 9d 23 79 09 93  3f
3731 : 68 e9 39 a0 92 88 6b 36  78
3739 : 73 c1 c2 88 82 fc 10 f8  91
3741 : a2 42 9b 64 ce 55 0a cb  cf
3749 : 50 ce 3f 12 29 20 2b 50  f3
3751 : ed bf 11 a5 d7 24 8d c4  75
3759 : 5d 52 e3 d1 49 06 b6 a8  03
3761 : 73 b3 b0 55 d2 2c d9 0a  8f
3769 : 2a e2 74 9d 27 34 19 91  71
3771 : 0d 66 ce 7c 3f 94 41 a4  db
3779 : 16 71 21 4d b2 67 c6 42  40
3781 : b2 d4 2e 1f 89 13 dd 26  02
3789 : 1d b7 e2 84 bc d2 e8 db  89
3791 : ae 49 14 22 f9 8b 8f 1f  a6
3799 : 7c d0 30 70 43 e3 bb 0e  f6
37a1 : f6 f0 da 34 45 0c da 14  95
37a9 : 45 0c e2 93 46 4d 0a 1f  55
37b1 : f5 76 ab 7a a9 72 b2 3d  8f
37b9 : cc bb e9 74 d6 4a 4f 64  31
37c1 : bb 43 17 ea d0 2c 2b 4d  f7
37c9 : 90 7b a6 38 f0 d1 f0 5c  e2
37d1 : f4 10 f5 af 03 4d 7e d2  7b
37d9 : 6a 9b 40 f7 40 8d c5 96  d5
37e1 : 82 cf 62 7d d6 c4 fb 3c  8f
37e9 : d1 f7 59 a3 ec f7 e7 dd  6a
37f1 : 6f cf 74 99 c5 23 28 4f  4d
37f9 : b4 11 a3 e1 31 d8 9e 3d  2a
3801 : 1a 73 bb 13 c9 c4 9d cd  fb
3809 : 1e 49 79 dd f9 e4 b3 4c  14
3811 : e2 93 40 aa 4f 7a 51 52  d5
3819 : 0b c8 8f 76 91 52 0c ec  f1
3821 : 47 3d 23 8a 4d a2 6f b8  3a
3829 : 83 49 0a 7b 50 ce 3f 10  db
3831 : d2 42 f2 fd f6 82 8e 9d  9a
3839 : 92 48 5b 5f ca e8 0d c4  64
3841 : ba d8 9e 6b 39 24 c9 67  27
3849 : 75 bf 3c d2 f7 59 a3 d1  84
3851 : fc cb 92 47 92 ca 7c 3c  aa
3859 : 27 e2 65 ad 5c 78 d2 42  5a
3861 : 4a ed 0b db 59 7f 1e 9c  23
3869 : 5b 5b 21 ff a9 a7 26 36  97
3871 : d4 3f 7e 7a 7f a9 3f cd  b2
3879 : 1c db 0a 03 dc 4f 22 92  5c
3881 : 1b 88 14 64 65 aa a3 52  51
3889 : 2e 3d 1f c8 cb 02 8f fa  38
3891 : 23 59 47 f2 a2 92 1f b9  40
3899 : ea 1d 78 08 62 c0 42 6a  3b
38a1 : 01 39 68 2d 01 d2 66 b3  a6
38a9 : 8f a1 b2 e1 a3 7e 71 48  76
38b1 : c8 d2 e1 99 10 d6 72 48  a0
38b9 : 85 9c 43 32 67 3e d1 a2  98
38c1 : 15 96 a6 e1 a3 98 b1 64  96
38c9 : fb 40 c6 24 3e 67 0c 77  59
38d1 : e8 6a 6f c3 e3 bb d9 ef  a6
38d9 : 4f e6 70 d1 a7 93 ed 6d  9b
38e1 : 3d 23 e1 ed 5d 8f 04 34  b1
38e9 : b9 da 0d 01 ec 3d 23 dc  72
38f1 : 67 11 d4 53 cd 21 ab bb  8c
38f9 : 3b 9c 43 21 f6 3d 7e ee  a8
3901 : 0f 8f 21 ec ec 17 bb 3f  b3
3909 : d2 7a 9b 83 d8 6e 0f a7  fc
3911 : 0e 83 66 1e d9 fa 56 52  b2
3919 : 4c 04 c6 9e 2a 40 d5 16  15
3921 : 1f 49 ea 7e a3 d8 7a 87  69
3929 : a9 74 78 77 b7 86 d1 8e  2d
3931 : 7e a6 63 46 e6 d4 da 11  47
3939 : cf d4 cc 68 5c d1 54 1c  90
3941 : 8a 1d cb c3 b7 8a 35 62  2e
3949 : 74 f2 89 31 50 4c 4e 76  4c
3951 : d4 ac 54 2b 15 8b b3 dc  2c
```

```
3959 : 9e 9b 9f 5e 63 99 ae 8e  53
3961 : 91 3d 23 8a 46 46 97 1a  d4
3969 : 48 5e da ae 76 ca 8a cb  ec
3971 : 86 39 f3 8f 0f fc c1 7d  5d
3979 : 66 17 d9 67 26 4b 30 f0  2d
3981 : 9f 88 d2 e1 25 10 d6 61  46
3989 : f1 44 0b da 12 4c 99 96  d2
3991 : 97 4b 16 dc 2f 68 68 cb  5e
3999 : 04 73 c6 f5 c9 32 6e 61  72
39a1 : 3d 5a 33 c6 7c b5 98 23  cf
39a9 : 6c 1c e6 c3 c0 43 dd bf  72
39b1 : 09 28 86 b3 43 e2 88 10  74
39b9 : b3 d6 5e 92 39 8b 17 7d  08
39c1 : 32 8b 73 ad c3 51 79 29  4a
39c9 : 69 33 2b 32 33 e5 7d 53  dc
39d1 : 32 47 7c 19 d8 81 3d 9e  b5
39d9 : 65 fc 50 77 6a 8d c1 df  99
39e1 : 60 7f 9c b8 b4 e1 7b 42  0c
39e9 : 1a f6 d3 6c ad ef d0 ce  3c
39f1 : c4 39 30 ff 39 67 6c 08  ee
39f9 : 64 3e c7 b0 97 ab a1 e9  b5
3a01 : 07 53 05 87 96 25 0a 93  c6
3a09 : 0e f4 4a 09 23 b3 b0 48  68
3a11 : 56 93 59 fd 71 52 5d d0  08
3a19 : 1c f9 0c 87 d8 f5 0f ce  3d
3a21 : 69 b2 c7 4a 0a a7 a1 ea  59
3a29 : e5 f3 2a 19 53 cd 89 9c  b9
3a31 : f4 18 30 50 f8 79 ce 7b  d5
3a39 : 93 1d 04 3e 1e 71 2c c1  c5
3a41 : 8b f1 1a 5d 01 65 ba 7d  18
3a49 : 33 65 ea a6 6b a6 7c 85  a7
3a51 : b3 03 97 00 dc 34 70 96  ca
3a59 : f2 43 97 02 cb 86 8e 62  03
3a61 : c5 93 ed 06 bd b4 3e 67  75
3a69 : 0c 77 e9 ad 4a 6f 49 13  cc
3a71 : d6 d8 a1 b2 80 e6 66 63  12
3a79 : d5 9a e0 94 9f 87 04 44  35
3a81 : f5 b5 df 52 46 97 1a 48  ad
3a89 : 5e da ae 49 1a 26 56 a7  a5
3a91 : 7f 72 59 d9 70 c6 92 0c  7a
3a99 : ed 43 59 9e 0f d1 7e 69  9e
3aa1 : 70 cc 88 6b 30 b3 88 b1  2d
3aa9 : 42 82 d8 dc 00 63 10 cc  f3
3ab1 : 01 8c 40 df 67 a2 ea a4  85
3ab9 : f8 00 c6 20 4b 67 87 58  26
3ac1 : 1a 6b f3 8b 50 98 7e 2c  1b
3ac9 : f0 ec 42 f2 24 dd 9c 7d  bd
3ad1 : 11 a5 c3 32 21 ac dc 67  a6
3ad9 : 64 a8 c2 ce 20 5e 44 74  0b
3ae1 : 9a cd a1 d9 94 f5 c0 8e  1e
3ae9 : 8d 75 5a 29 21 ff ec b0  14
3af1 : f7 5f 88 3e db f4 2f ac  fd
3af9 : 82 1e 0c 39 8e d1 c7 93  72
3b01 : c7 59 b4 39 3e d0 ed 8e  08
3b09 : 9b 98 e1 ac 5d f7 69 0f  58
3b11 : 77 2c d7 4c e9 56 ee 87  3a
3b19 : 2e 02 1f 86 5a a5 95 54  b3
3b21 : 73 d2 bb 91 9f 2a b5 41  c3
3b29 : 95 ca 11 b7 39 5e 51 96  58
3b31 : ca d6 b2 9e 72 b3 9d cb  ba
3b39 : 95 58 0a c3 bc 15 80 f4  56
3b41 : 73 51 ef 55 9d a0 2d 84  a0
3b49 : c8 ac 2a 85 60 3d 1a 85  06
3b51 : b0 9c 19 e4 b4 4f ae 65  7d
3b59 : 5f 6e 6c 96 eb 91 57 3b  fc
3b61 : 99 85 b3 e9 9a d1 76 f0  db
3b69 : 75 8d c3 84 d8 94 d4 89  bf
3b71 : 9c 29 43 8f 94 39 29 43  a3
3b79 : 9d ca 1d 3f 28 74 f4 a1  e8
3b81 : d5 92 87 37 94 3a 49 78  99
3b89 : 70 9b 2c 59 bb ab 7a 70  e1
3b91 : e5 c0 68 be 4a 38 6b db  94
3b99 : 34 a8 4c d6 6b 93 1b 33  35
3ba1 : 7a ab 9d b2 a7 b2 f4 87  a1
3ba9 : 3d 27 da 5d 36 3d 42 17  61
3bb1 : b4 2b cd 2e 19 91 0e fb  82
3bb9 : 02 ce 21 da c0 63 10 ee  0b
3bc1 : e0 31 88 d1 06 31 1a 34  51
3bc9 : 83 31 eb 9d a3 27 30 ef  a7
3bd1 : a0 1e 5a ac 53 7a e0 c6  c7
3bd9 : 21 b9 99 2d 41 8c 43 b9  dc
3be1 : 00 c6 24 9f ae 0c 62 43  9d
3be9 : da f2 19 5f 85 e4 47 46  98
3bf1 : f5 f2 79 04 f5 d9 1f 2b  3f
3bf9 : cc 5d 32 cb 84 6e 62 ea  95
3c01 : 26 54 c7 3c 07 96 a7 4f  2d
3c09 : 03 18 87 5b 03 18 86 99  a4
3c11 : 06 31 0d ea 83 18 81 31  b2
3c19 : 9e eb 87 d7 6d 28 2b 06  5a
3c21 : 7e c8 79 f6 c8 b8 5c d9  b8
3c29 : 1e a9 0c 1e 7d b1 3f 00  45
3c31 : f3 ed 89 eb 2c 50 f3 ed  ec
3c39 : ad 3f 08 74 c6 1d fb ad  b7
3c41 : 07 5a 64 93 2c b5 f2 be  bb
3c49 : aa 1f 85 ab 83 e4 aa c8  f5
3c51 : c9 0a 4c b4 3d db f3 1e  88
3c59 : 4e 42 ed 83 29 f4 5c 20  a0
3c61 : da 7d 84 f9 f3 33 c7 92  77
3c69 : e1 ce 12 be f9 c5 22 17  92
3c71 : 5d 61 46 67 a1 56 e0 e3  15
3c79 : de e8 86 79 99 dd 5b 8a  a7
3c81 : 33 3c 1a 83 69 ec 50 26  55
3c89 : b0 e9 f2 7a 01 c9 cb b7  37
3c91 : bc 60 2d 42 eb 94 64 a9  59
3c99 : bd 01 a2 86 3c f5 bd e7  8a
3ca1 : 11 80 b5 1b 49 b2 c3 da  b2
3ca9 : cf 8f 04 39 39 76 f7 9c  c8
3cb1 : 46 02 d4 6d 1c 16 1e d6  74
3cb9 : 7c 78 21 bd bb b2 dd 6d  15
3cc1 : 70 a2 f9 d6 7d f8 b7 5d  15
3cc9 : 61 0f ca 35 cf 17 cc 1b  2a
3cd1 : 33 6a 76 9d 61 1b 23 69  d9
3cd9 : df 89 0f a6 7c e3 e5 0e  b0
3ce1 : 76 43 e9 b5 83 fa 5f 74  a0
3ce9 : 7c 7e b0 a7 7b f1 0d de  ff
3cf1 : d4 a8 af 98 e8 af 36 15  27
3cf9 : d7 7e 2c 58 75 06 b8 ba  05
3d01 : eb 0a ba dc 51 45 52 4b  5b
3d09 : 89 8c 29 54 1a 62 9c 5d  8f
3d11 : a9 46 ee 57 a2 6e 8d 45  e2
3d19 : 75 b8 ae bf 40 b9 97 c8  50
3d21 : f8 13 17 c6 b8 7d 84 a4  14
3d29 : f9 f4 13 c7 9c f4 9c 75  29
3d31 : 3f 61 e3 a7 73 80 63 d9  8b
3d39 : ff d1 d3 c3 fe 2a cf b9  82
3d41 : 3c 39 39 76 f7 8c 05 a8  80
3d49 : 5d 72 8c 94 61 c0 38 f1  76
3d51 : f7 d6 f7 9c 46 02 d4 6d  e7
3d59 : 26 cb 0f 6a ec 78 26 f2  87
3d61 : fc be f7 9e e3 b7 34 6e  38
3d69 : f7 4b a5 d2 e2 ce ad a3  6c
3d71 : dd 6d 74 6b b0 a2 fa f3  83
3d79 : 53 bf a0 48 ce b9 c0 39  0d
3d81 : 22 7e 40 e4 e5 db de 71  2b
3d89 : 18 0b 51 69 1e 18 f2 d6  c4
3d91 : 97 b7 bc e2 30 16 a3 68  a2
3d99 : 51 43 1d d8 51 e8 d5 76  8f
3da1 : 6a 8e 2f 98 36 e1 95 97  29
3da9 : c5 d5 a3 3f 39 09 0c 9a  6b
3db1 : 2b 2c 0b 3f d0 22 4f 84  01
3db9 : 26 d6 7d 61 04 26 d6 7d  9e
3dc1 : f8 82 13 6b 3b 71 62 d9  a9
3dc9 : 95 20 84 da 1d 34 f8 f2  28
3dd1 : ce 2b ec 0b af d0 2a 64  c4
3dd9 : 11 ef 06 cb ae c0 ac 5c  39
3de1 : 7e 35 0a c5 c6 70 9e 44  28
3de9 : a5 6d 85 bf fe b9 31 b7  90
3df1 : 0e 33 93 4c 0a c5 9d b3  b4
3df9 : 0c 71 ae d9 32 92 89 a2  c8
3e01 : 1b 8a c5 d3 d1 51 60 56  23
3e09 : 2e 33 aa 15 8c ed 9d 54  75
3e11 : ff 90 a1 58 e9 74 ae 23  0f
3e19 : b5 f2 0a a1 58 9b 70 e1  e6
3e21 : 19 ff 21 42 b1 d2 e9 5c  dc
3e29 : 47 6b e4 15 42 b1 71 f3  61
3e31 : 4d f4 e3 9f 24 cd 42 b1  02
3e39 : 7c 69 26 6a 15 8a 1f d6  11
3e41 : 22 18 91 1f b7 59 fb 7b  e5
3e49 : 77 ed e9 df b7 9f 38 bc  00
3e51 : 5e 9f a3 d3 0f ae 24 54  82
3e59 : 83 67 a1 0e c5 0c b7 4f  f4
3e61 : 3e 9b 0c 21 a9 99 60 71  60
3e69 : 51 e3 c1 7e dc 1f b7 66  5e
3e71 : fd b9 ef db 89 4c 21 9d  7d
3e79 : 56 d6 b6 c1 4a db 04 e5  7f
3e81 : 2b 6c 2d eb 58 66 58 29  18
3e89 : 84 36 15 b6 0e 3c 05 2b  72
3e91 : 6c 13 7e de f9 fb 7a b7  db
3e99 : ed db bf 6e ed 72 43 ad  0c
3ea1 : 60 56 2e 1b a3 3e 6e 9d  3c
3ea9 : 3c 5f 8c e4 83 51 78 87  08
3eb1 : 06 24 13 8a 43 20 14 2f  c3
3eb9 : c0 b2 85 63 6f 6d 29 5b  5e
3ec1 : 61 6f f3 39 30 55 db 42  9f
3ec9 : 92 66 a1 58 c7 76 ad 5b  9f
3ed1 : 96 4e 33 aa 15 8b e3 1c  26
3ed9 : ce 50 ac 6d e9 ea 32 84  70
3ee1 : 3b f3 b4 c8 78 31 10 be  2b
3ee9 : a1 94 2b 1f 1a da 50 ac  96
3ef1 : 6d ed 98 63 8d 76 c9 94  c4
3ef9 : 94 4d 10 dc 56 2e 9e 8a  3a
3f01 : 8b 02 b1 71 f8 d4 2b 15  35
3f09 : ed 17 b4 99 2b 19 d0 d0  42
3f11 : c1 99 2b 1f 12 93 43 06  24
3f19 : 64 ac 7c 48 fa 18 33 25  83
3f21 : 63 e2 54 68 49 64 70 35  fb
3f29 : 2b 15 0a c6 68 d6 49 f1  80
3f31 : 0d b5 8e 94 a5 6d 82 66  ec
3f39 : f5 88 c0 df 81 38 09 88  ad
3f41 : b4 e5 db 96 71 26 7d eb  c8
3f49 : 9e f8 99 a5 5b 9c d8 9e  ba
3f51 : 45 41 bb ea ed ed b6 29  fe
3f59 : b3 55 da 0b 66 54 f3 1d  e2
3f61 : f1 33 52 be 40 e0 a3 a1  35
3f69 : 37 91 1e b2 4a 38 50 c0  70
3f71 : 38 b1 a3 0c 50 99 f5 13  bc
3f79 : 9b ba 67 6c bd fb bd 28  dc
3f81 : 58 db 34 71 9c 92 8e bb  12
3f89 : 56 c8 fc 99 e4 28 60 06  d3
3f91 : bf 2d 6e e2 be a8 6c a0  03
3f99 : 2c 7f 04 a1 9b b6 8d 52  04
3fa1 : 3b 54 d6 94 0c bb 64 34  e7
3fa9 : d7 f4 bd b8 e7 ca f3 66  f2
3fb1 : a6 f3 91 e7 97 70 86 ec  a3
3fb9 : 99 c7 1b ca 56 a8 e5 42  1d
3fc1 : 43 ca f8 03 a9 ce 35 98  1f
3fc9 : f0 83 9c f4 3e d2 84 26  19
3fd1 : 01 6b c3 7d 71 8a d0 ef  b7
3fd9 : de bc e2 a6 66 43 a2 43  34
3fe1 : 9c d9 62 c9 f6 97 55 8f  dc
3fe9 : ca 34 b8 66 44 3b ec 0b  b0
3ff1 : 38 87 64 4a 49 f6 81 8c  bb
3ff9 : 48 7c ce 18 ef c2 f2 22  5b
4001 : 1e 13 f1 18 4e 2f 9d 6f  dc
4009 : f5 49 6f c4 96 fe b1 2a  94
4011 : 59 44 e0 85 e4 44 3f 2e  3f
4019 : 86 52 b6 c3 c6 49 0b da  87
4021 : 14 25 2d 43 67 a1 37 4d  77
4029 : 11 26 87 e5 d0 cc a8 7d  fd
4031 : 1c f0 ef 5f 7b 94 73 c8  69
4039 : ff aa 34 87 24 76 f0 86  52
4041 : 43 ec 7b 06 0e bb f4 1e  69
4049 : c0 56 0e 19 d2 e1 4d 4e  e9
4051 : e7 2b 98 a5 7c 34 a3 0f  bf
4059 : 2d 3a 83 70 2c 66 4c c2  3f
4061 : c7 d5 4e 57 80 33 e5 58  7b
4069 : 26 ab 6e bf 4b 9a 07 d7  4e
4071 : 0d 2d 71 6f a8 35 43 0e  bc
4079 : 9d 70 5b d7 06 15 70 7f  ea
4081 : 36 d9 fd 71 56 55 b5 01  3a
4089 : f6 13 94 09 58 06 9a 40  f0
4091 : d8 7a fe b7 79 51 1d 58  a4
4099 : eb 57 3a 73 eb cc 73 16  4c
40a1 : ed c9 fa ec a7 ec ee 12  91
40a9 : b4 f5 f6 97 09 54 b3 80  0b
40b1 : 9f e4 83 ee 5e 79 97 9a  c6
40b9 : 57 e2 b4 c1 ad c1 b7 c1  32
40c1 : f7 5f 8b e8 81 1a 04 4b  f7
40c9 : ff 7e 07 2d f8 bd 1b c3  e0
40d1 : 5d c4 cb 38 b8 bb 43 cb  98
40d9 : b4 29 fb 4f d6 7c 31 dd  dc
40e1 : e2 9d 7c a4 23 e8 4b 7d  e7
40e9 : fc 8b 7d ec 16 fa 35 51  58
40f1 : 6f 86 7d 28 13 d6 1d 68  35
40f9 : e4 96 2c c6 32 ce 83 fe  b2
4101 : ec 50 08 1a 09 ab 2f 59  b8
4109 : d0 77 2f 0e 6e a7 de 79  b5
4111 : ce d0 83 5c 9c cd 31 67  7f
4119 : 31 73 ee 93 3b 78 a3 5b  ef
4121 : 05 1a b1 50 b6 07 37 52  cf
4129 : b6 c1 4a d6 2a 67 2c 14  e4
4131 : ce 31 54 b4 09 58 50 24  20
4139 : c6 81 d7 52 35 01 f6 07  45
4141 : 56 28 56 27 6e d4 3d 89  bb
4149 : dc 62 08 ef 54 85 3b aa  0a
4151 : a3 bb 03 aa d4 9f b0 43  7b
4159 : ff 1b 29 52 79 cd c9 2e  04
4161 : 9b fa c8 24 e1 b9 ed 4c  6c
4169 : 78 e9 93 88 e6 5e 62 cd  52
4171 : 33 4b ae 3c 6d 58 6f 4f  73
4179 : 5e 71 54 de 9e d1 32 5d  7d
4181 : 19 94 fb 22 32 4d 42 1e  fa
4189 : 9b 29 e9 9f 8c fe 04 c1  7b
4191 : f4 1f 19 fb 39 83 d3 65  a4
4199 : 3d 33 d1 3f 81 f7 9f 41  a5
41a1 : d1 3f 67 f7 9e 9f 5f 5c  08
41a9 : e9 25 f0 97 e8 17 9d bc  8b
41b1 : 51 ab 1e 03 56 31 f5 00  87
41b9 : 7d 70 d2 d7 1f 7c 1e 3d  e7
41c1 : f8 73 9b 0f 30 30 ab 83  76
41c9 : a7 5c 0e ac 75 c7 ad a0  45
41d1 : ee 6e 4f 9a 70 24 69 18  1b
41d9 : 15 8b 8b 69 fe 0a 85 62  df
41e1 : c3 72 7a 8d 58 c2 71 8e  ac
41e9 : 8b 57 07 a3 3c 19 bb c4  5b
41f1 : 2e af 02 b1 b0 76 a8 50  b0
41f9 : fe d7 90 9c 1e 8c f0 66  71
4201 : ef 10 fe d7 90 9c 33 45  f8
4209 : 20 ac 6c c3 72 7a 8d 58  f5
4211 : b4 c5 69 34 d2 ac 9d c7  21
4219 : 31 c9 f9 10 f6 1f 88 eb  12
4221 : 12 19 0f b1 eb f7 70 58  ab
4229 : 09 8b 81 13 e0 61 3c c7  54
4231 : 2b 51 bf da 1e a4 f3 16  53
4239 : 15 8c 77 fd 23 d4 ac 62  82
4241 : 76 89 4c e3 16 1b 93 d4  3e
4249 : 6a c4 e6 ca 4f 31 52 b1  53
4251 : 8a 1b 1c eb 0d c1 ea 4f  16
4259 : 31 61 fa 4f 52 b1 89 da  72
4261 : 25 33 8c 4e c7 d6 6e 4f  98
4269 : 53 72 79 cd 94 9e 62 a5  20
4271 : 63 19 9d 01 ee 71 53 9a  e5
4279 : 61 fa 4f e7 e2 c2 c9 86  a1
```

Listing 1. Die zum Brennen benötigte Software für das EX-SMON-KERNEL. Bitte mit dem MSE eingeben

```
4281 : a6 d0 fd 06 49 54 e9 1e  eb
4289 : 9e af 9e dd a3 8c ea f5  98
4291 : f0 97 e8 17 cb ce bb 13  b2
4299 : 86 68 a5 ae d5 25 ae b8  45
42a1 : 48 1d 58 9c 25 1f 1e 83  ec
42a9 : 1a e0 c4 b1 54 88 70 de  a4
42b1 : 9d d1 51 ad 83 0a db 07  47
42b9 : 17 d2 5a 25 2b 6c 10 a0  0c
42c1 : a1 d3 a1 fc db 21 d3 ae  c7
42c9 : 42 de b9 0c 2a e4 3c aa  7a
42d1 : e4 1d 5e 21 67 11 0f 8a  50
42d9 : 22 1d 5b c4 3f 5d e2 1f  a2
42e1 : 24 14 38 f7 88 5e d0 a0  1c
42e9 : c6 22 19 d8 88 5e 44 43  35
42f1 : 32 22 12 51 10 ac bc 40  bd
42f9 : b7 88 51 57 6b bc a4 b5  ce
4301 : da 84 b5 dc 84 b5 c0 96  4c
4309 : bb 0d 2d 74 da 5a ef 2d  bf
4311 : 24 3d 4b b4 18 e5 a0 f6  5e
4319 : fd 06 37 fa ea 44 b5 dc  a8
4321 : f4 b5 d2 89 21 f4 df a1  52
4329 : 3f 62 85 3d 8a 05 b1 fc  34
4331 : e6 f2 be 0d 3d e6 c3 3c  74
4339 : bd b1 40 4d ef a4 82 b6  24
```

Listing 1. Die zum Brennen benötigte Software für das EX-SMON-KERNEL. Bitte mit dem MSE eingeben (Schluß)

```
Name : dos v3                0801 26a9
--------------------------------------
0801 : 0e 08 ca a8 9e 32 30 36  84
0809 : 35 20 46 43 43 00 00 00  7d
0811 : a0 00 b9 69 07 99 00 cd  26
0819 : b9 69 08 99 00 ce b9 69  ec
0821 : 09 99 00 cf c8 d0 eb 4c  4c
0829 : c2 cd 78 a0 ff 84 fb a9  6b
0831 : c6 85 fc a9 36 85 01 8d  dd
0839 : 20 d0 c8 a5 2d d0 02 c6  97
0841 : 2e c6 2d a6 2e e0 0a d0  a6
0849 : 04 c9 66 f0 0f b1 2d 91  40
0851 : fb a5 fb d0 02 c6 fc c6  10
0859 : fb 4c d3 cd a2 08 a9 01  3c
0861 : 86 2e 85 2d 84 ff 20 50  6f
0869 : ce c9 f3 d0 27 20 50 ce  85
0871 : aa 86 fa c9 04 b0 04 a9  7f
0879 : f3 d0 03 20 50 ce a0 00  97
0881 : 91 2d c8 c6 fa d0 f9 98  03
0889 : 18 65 2d 85 2d 90 02 e6  7d
0891 : 2e 4c 34 ce a0 00 91 2d  77
0899 : e6 2d f0 f3 a9 2d a2 28  af
08a1 : e4 2e d0 c2 c5 2d d0 be  af
08a9 : a9 37 85 01 a9 fe 8d 20  78
08b1 : d0 58 20 59 a6 4c ae a7  b7
08b9 : a2 ff 86 f7 86 f8 e8 a9  22
08c1 : 01 85 fe a9 7f 85 fd c6  23
08c9 : ff 10 10 e6 fb d0 02 e6  cd
08d1 : fc a9 07 85 ff a0 00 b1  7d
08d9 : fb 85 f9 06 f9 b0 0a a4  6d
08e1 : fe a5 fd 39 f7 00 99 f7  2e
08e9 : 00 8a 0a a8 a5 f7 38 f9  b5
08f1 : e2 ce a5 f8 f9 e3 ce 90  de
08f9 : 0e e0 0c f0 0a e8 38 66  2e
0901 : fd b0 c4 c6 fe f0 bc 8a  e0
0909 : f0 0f a5 f7 38 f9 e0 ce  5e
0911 : 85 f7 a5 f8 f9 e1 ce 85  0f
0919 : f8 a4 fe f0 07 a5 f8 85  ce
0921 : f7 88 84 f8 a5 fd 4a 90  31
0929 : 07 46 f8 66 f7 4c be ce  d9
0931 : bd d2 ce 65 f7 a8 b9 00  63
0939 : cf 60 00 00 00 00 02 05  4b
0941 : 11 29 63 9d ce ed f9 fd  b4
0949 : fd fd 00 00 00 00 00 00  45
0951 : 00 20 00 38 00 68 00 98  dd
0959 : 00 d2 00 ef 40 fb 20 ff  25
0961 : e0 ff 00 00 00 00 00 00  41
0969 : 20 85 a9 00 d0 a5 02 4c  91
0971 : 01 f0 c8 8d 18 60 1c 29  15
0979 : ad 0a 03 05 91 fe 10 06  ea
0981 : 30 a0 ff 80 b1 99 aa a6  f1
0989 : 50 e6 f9 04 e9 48 86 a2  8b
0991 : ca 45 88 90 68 fa 4a 09  cc
0999 : 69 81 98 c9 52 8a e8 cf  df
09a1 : ea 2e 95 c5 de f1 f7 94  47
09a9 : c1 07 08 53 54 7f b8 c6  0c
09b1 : d5 84 0f 6d 6f a4 bd c0  ce
09b9 : 31 b0 f8 0c 55 f3 2c 38  18
09c1 : 82 d4 dd e0 f6 4f 9d b5  0c
09c9 : ef f2 44 59 87 df f5 12  e1
09d1 : 1e 49 ae 36 43 a8 b9 e7  36
09d9 : fb 0d 4e 51 70 d1 3e cd  43
09e1 : 3f d6 e5 0e 32 34 89 e2  77
09e9 : fd 75 ce 0b 24 41 57 65  2a
09f1 : 78 93 9c e1 fc 40 4b ee  73
09f9 : 1d 2f 4d 58 d2 d3 19 2a  91
0a01 : 56 6c 72 8e bb f4 11 13  c9
0a09 : 22 25 26 39 5e 62 63 6a  ca
0a11 : 83 ed 16 28 3d 5b 7a 17  dc
0a19 : 1a 23 3a 71 97 d7 1b 21  68
0a21 : 42 5a 66 76 a1 a7 c7 eb  47
0a29 : 15 1f 5f 67 73 79 7c b7  f7
0a31 : be c4 d9 e3 27 2b 47 9a  62
0a39 : 9e ac cb cc 35 64 6e 8f  09
0a41 : ba bf da 14 2d 37 74 7d  6d
0a49 : 9b ab db dc 33 3b 3c 46  d7
0a51 : 5d 61 77 7b 8b 9f a3 b2  56
0a59 : b4 b6 d8 e4 ec b3 bc c2  20
0a61 : c3 5c 7e 92 af e2 58 d7  67
0a69 : d8 7d cd eb bd 65 f9 4a  74
0a71 : 52 62 bc b1 1f 86 e1 62  cc
0a79 : 7e 87 9e 50 fb 17 1e 61  20
0a81 : 11 79 ce 1e a5 c6 90 3f  17
0a89 : 2f f9 a1 c0 e6 94 48 01  6b
0a91 : 9b 92 11 be a9 13 5e 51  e1
0a99 : 0a 19 79 06 f5 a6 e8 8f  a6
0aa1 : 4b 50 47 c1 e9 e4 87 4b  99
0aa9 : 6d 9d 02 06 89 e4 d9 08  5d
0ab1 : ac 1f 83 2a 16 39 0b 53  11
0ab9 : 5e b4 dc 2c 7a 9a f4 3f  fd
0ac1 : 17 30 3a db 60 a9 ab 0f  1a
0ac9 : b9 a6 87 e4 87 87 56 1a  96
0ad1 : 9a 62 1e 09 e4 07 ed c9  17
0ad9 : 37 14 38 f5 68 40 b7 0d  69
0ae1 : 4d 30 7e dc 93 3a 0e 3d  3f
0ae9 : 58 40 b7 0a 9a b3 9f 25  a0
0af1 : 3c 1a 9a 60 e3 d5 86 a6  41
0af9 : 99 60 6d cc 0e 56 99 a4  fa
0b01 : 13 86 82 07 fa c0 4c d3  67
0b09 : d7 98 fa fa f2 a4 d1 e1  aa
0b11 : f4 37 98 ff 4d 79 4c d8  4a
0b19 : 80 87 a7 93 97 00 cd 43  f0
0b21 : f9 65 2c 47 81 0b 24 df  82
0b29 : 59 c2 1f 2e 51 d8 29 6c  ca
0b31 : 96 20 04 2c 94 9b 13 68  a1
0b39 : a7 68 f0 c4 47 28 2c b9  c3
0b41 : 8c f6 8f 49 aa 09 d7 a1  eb
0b49 : ff 71 18 7c 66 c7 60 a5  08
0b51 : fb 96 38 64 02 ff 54 d8  55
0b59 : 89 1f 06 6e 4a 1f e9 94  30
0b61 : 40 08 59 26 68 42 88 87  8a
0b69 : cb 94 86 22 39 4b 13 9c  d8
0b71 : 1e 9e 48 7d cd 34 f7 3b  75
0b79 : 57 c7 3e a7 dc 58 3e e6  90
0b81 : 99 8f cb ab e3 9f 53 ee  b0
0b89 : 2c 47 c1 0b 24 df a4 3a  73
0b91 : 5b 6c e8 1f 4e 89 e9 77  08
0b99 : f9 70 0c e8 3f 96 50 74  bd
0ba1 : b6 de dc 08 1b 73 c9 bf  f2
0ba9 : 81 a0 5c e8 1f 4e 89 e9  0d
0bb1 : 22 b1 28 20 68 15 60 36  d7
0bb9 : 98 3a 5b 63 85 0f 2f 99  72
0bc1 : 0b 1c 9b 2c dd b9 e4 a0  c7
0bc9 : a9 d0 24 30 4e be 16 3e  99
0bd1 : 75 78 e9 6d b4 b4 47 9a  ee
0bd9 : 31 1e 8e 97 3b dd 19 90  d8
0be1 : a5 c8 85 8f 55 a5 9a eb  02
0be9 : a5 e7 06 a6 98 3f d3 99  de
0bf1 : 35 63 0b 1f 37 44 f4 35  52
0bf9 : 34 c1 a9 a6 23 d1 d2 87  68
0c01 : fa 73 30 1a 85 2e 44 2c  37
0c09 : 7c dd 13 d6 0f b2 ae f3  3d
0c11 : 4a fe b8 db 29 9b e3 e6  51
0c19 : a3 5b 4c 9b 11 1b 69 95  ab
0c21 : 56 a1 c6 58 3a b5 63 6e  c0
0c29 : 48 c3 5c f1 a1 a3 4c cc  aa
0c31 : bb f0 5f d7 25 83 43 3b  29
0c39 : 55 6e 94 f0 db c2 c7 ce  99
0c41 : dc 07 ab a6 78 5b 8a e7  bd
0c49 : f5 c6 e2 26 fe bb 5d 33  c8
0c51 : 71 ba a1 42 3d 59 3e 20  a8
0c59 : 98 88 1f f5 ff 8c a2 78  9c
0c61 : f8 e1 ef 1a 69 b1 36 a0  c7
0c69 : 9d 79 1e 06 6e 48 78 1b  4c
0c71 : 64 9b 13 f0 82 75 f9 df  01
0c79 : f2 c2 23 68 c9 e7 d4 e7  a1
0c81 : 32 d4 fa 9e ec 2c 72 6f  89
0c89 : ea 74 19 8c 7a 4e 71 a3  ac
0c91 : a2 97 12 df dd 9a b0 26  41
0c99 : d1 4e d1 e4 28 e4 06 f3  4c
0ca1 : 86 18 35 70 ec 48 f8 3d  fe
0ca9 : 3c 94 3c 3e 60 70 b8 04  7b
0cb1 : 7c 10 b2 52 6a 87 87 cc  c7
0cb9 : 42 ba bc 39 7c 03 bd 40  06
0cc1 : 4f 22 16 3e 77 bb 8d bb  72
0cc9 : c8 c0 6a 81 fc 05 79 1e  d6
0cd1 : a4 d3 59 f4 71 82 7b 29  bf
0cd9 : 0c 6e 61 36 80 9d 79 d5  c2
0ce1 : c3 bf 3a a4 69 00 dc 03  b7
0ce9 : 22 b0 56 64 86 15 59 cf  9c
0cf1 : 92 de 07 5b 80 67 41 a9  bb
0cf9 : a6 43 f3 75 43 f3 75 5e  53
0d01 : 15 fe fc 5f 26 ff 7e 34  85
0d09 : 08 7f fc c1 b7 ab 78 57  b1
0d11 : fb fa a7 c9 22 b2 6f f7  12
0d19 : f5 54 0b 07 d9 56 f0 f2  d6
0d21 : 15 bd 72 66 d8 84 6f aa  43
0d29 : a6 bf 5c 47 ad eb 82 37  61
0d31 : d5 35 de 1a 31 0b 53 5e  11
0d39 : b4 2c 7c ea fc 64 b6 4a  e2
0d41 : f0 bd d9 ae 21 9d 10 fc  95
0d49 : 5f be e1 69 f4 71 96 0e  fe
0d51 : b6 d8 3e f7 30 39 7c 00  c1
0d59 : fe f4 c7 6e 87 4a 75 b8  a3
0d61 : c9 21 46 64 85 1c 80 de  d2
0d69 : 70 c3 a7 9f 79 8f fe de  66
0d71 : ec ba 45 79 1e a4 d6 14  c5
0d79 : 4d d4 4d d1 9d 44 b7 23  df
0d81 : 91 dd 1d a5 56 7c f0 8f  29
0d89 : 56 75 bd 13 a9 a6 c4 a3  96
0d91 : 04 eb ce dd 63 6e f4 b8  e9
0d99 : 0d 63 dc aa 3d b3 ad f7  fc
0da1 : 72 47 e3 94 59 3b 59 3e  94
0da9 : 94 10 da 69 8e f4 8d 61  b3
0db1 : 0b 6b 2c c7 c4 99 dc 7c  fb
0db9 : 67 cd 4f 1b 70 cf 6c fa  6b
0dc1 : ad ac d5 b1 11 0c bc 85  df
0dc9 : 67 d1 85 8f 53 5e 1a 9a  32
0dd1 : c0 e0 e4 9d 32 6f 7c 3c  f7
0dd9 : 3a b0 d4 d3 01 b7 41 a9  41
0de1 : ac 0e 0e 48 70 38 08 64  d3
0de9 : 7d 41 a9 a6 03 6e 83 81  fb
0df1 : c0 0e bd 58 78 75 61 85  f6
0df9 : 56 73 e9 07 50 48 0d bd  5b
0e01 : 59 d4 12 0e 7d 20 26 34  e5
0e09 : d0 4c d0 36 7f f0 47 21  d9
0e11 : 1e ae 24 99 b0 53 26 6c  da
0e19 : 14 c9 9b 1a 09 b1 7c 95  77
0e21 : c9 d2 6a ad cd ee 04 c6  95
0e29 : 9a ab 75 34 f1 2c 00 98  2e
0e31 : d3 48 21 43 f7 d5 90 ba  bf
0e39 : e6 f5 c9 cd e3 47 45 bd  4f
0e41 : 70 76 aa c2 05 b8 77 9d  1e
0e49 : 00 de 70 c3 53 4c 18 35  af
0e51 : 7b 56 a8 70 b8 04 7c 1e  09
0e59 : 9e 49 21 82 75 e1 a9 ac  9b
0e61 : 0d 4d 30 56 64 87 2b 4d  b5
0e69 : 20 2b ea e1 63 e7 6e 35  af
0e71 : d3 70 48 e4 0e ab 21 43  74
0e79 : f7 d5 bc 3d fd 74 82 14  e7
0e81 : 13 c8 9b 56 7d 16 f5 cd  a6
0e89 : 27 e3 0e ff 5d 8d 1c f1  bc
0e91 : a4 39 f0 e9 5b d7 26 75  43
0e99 : 00 cc b7 ae 78 5a 16 f5  62
0ea1 : c9 93 bd 6f 5c 47 20 9e  4f
0ea9 : 40 43 ab 08 16 f0 b1 f3  0e
0eb1 : b7 12 03 ee 69 a1 63 d4  eb
0eb9 : ee 26 f5 7c 73 ea 7d c2  d1
0ec1 : 14 13 c8 78 53 9f 37 ae  0c
0ec9 : 4c ea 06 f5 c8 55 ff c0  83
0ed1 : 72 b4 d0 b1 c3 d5 d3 10  62
0ed9 : a0 9e 45 b5 25 b5 25 e6  33
0ee1 : fc ed 2b 7a e4 ce 99 6f  f8
0ee9 : 5c 1e 1d 58 40 b7 bc a4  a5
0ef1 : be 6b 81 fe e8 6a 69 8a  42
0ef9 : f3 33 ef 07 ad df fa 1c  61
0f01 : 5f af ab 31 c6 d1 df d1  67
0f09 : f8 e1 67 99 98 08 71 a8  e0
0f11 : fe d5 54 3f f0 e8 21 d1  75
0f19 : cc 59 4c 6c f2 36 94 dd  21
0f21 : 99 8a 3f e6 86 67 1c 35  2a
0f29 : 5b 3f f1 e0 79 54 9b ff  65
0f31 : 06 90 21 d1 e6 59 cc 6b  45
0f39 : 6c c3 33 86 c4 c8 6a b4  ca
0f41 : 45 dd fe f7 f8 6f ba da  df
0f49 : 4d 9d 9f 1b a9 bf 5a 91  d5
0f51 : ad c7 8b e3 84 c7 53 55  c0
0f59 : 65 67 d6 fc 3e 38 59 e9  a6
0f61 : 21 cc 0f 23 87 98 1e ba  3c
0f69 : 19 9a df bd ec fc 54 48  17
0f71 : 21 31 f6 b8 c8 05 9d 1f  69
0f79 : 1b 5b cf 0c ca 30 b2 98  e1
0f81 : d6 d9 e6 75 7f af 8e ea  32
0f89 : e3 c2 f1 3c 77 5a 54 7c  66
0f91 : 7b 20 b2 d6 d1 f1 ac f5  ef
0f99 : b2 26 3e 8d af 0f 30 35  3e
0fa1 : b2 38 db 3d 9c c7 df f4  7f
0fa9 : be 8d 55 1e ff b7 32 1f  0c
0fb1 : cf 0d 0f bf f1 ff b6 fb  b4
0fb9 : eb 98 a3 0d 1c cb 20 ec  f5
0fc1 : ff 00 bf fc 1b e7 ff ab  98
0fc9 : e8 a3 d5 76 66 1d 05 1c  63
0fd1 : c7 1b 47 e9 9e aa a3 a3  4a
0fd9 : cc ea b4 f5 28 f5 56 75  7d
0fe1 : 4d 6d b5 48 55 46 a6 87  8c
0fe9 : 31 c3 6d b8 d5 59 74 f8  5a
0ff1 : d6 7a d4 dd 70 f3 32 2c  bd
0ff9 : 2d b3 30 f6 7f 45 8f d1  ef
1001 : aa b2 fb dc 4e ac c6 b7  74
1009 : 32 8f e8 91 26 2d 23 48  58
1011 : 39 ce 85 f6 75 80 d0 4d  2b
1019 : b1 72 95 8d 15 c9 9d 13  57
1021 : f5 1a c4 8e a7 82 0c b9  58
1029 : 74 92 19 ec 1f 44 53 5c  e4
```

```
1031 : 91 ea 9d 83 e6 c3 dd 88  24
1039 : 1e b9 90 91 12 6c 3d d8  b5
1041 : 81 7b 5e 42 a9 14 f0 c5  ea
1049 : af 0f cd 90 42 8f f9 ca  23
1051 : 98 fb 67 5c 31 72 3e 3d  66
1059 : 15 ca 57 fc 93 be 43 2e  e1
1061 : 77 9a 67 30 bc 7b df b0  8e
1069 : fa 27 a9 7c 0c 5c 8e 3a  43
1071 : 1d c8 81 ef 1a 9e 9e 9c  9b
1079 : ce 82 c3 6d 4f 4f 4e 1e  0c
1081 : 06 da 8a 36 03 50 fa fa  f2
1089 : a8 6c 3d e8 24 2e 0e 51  62
1091 : 52 b0 56 7d 19 68 7f 94  7d
1099 : 4f f7 83 89 1a 74 3f d6  e6
10a1 : 25 23 17 33 ce db 9c 9a  f7
10a9 : c1 e8 57 90 af 0b 70 fa  d1
10b1 : e2 81 e1 65 5b c9 3a 51  08
10b9 : f2 61 07 ed 88 4c 56 e7  ef
10c1 : 26 cd b9 ea 6b 4f a4 07  6b
10c9 : 5b 02 62 b7 39 33 9f 07  6f
10d1 : ec 88 4c 56 e7 26 75 00  65
10d9 : 7e c8 93 34 2a cf db a9  0b
10e1 : aa 3e 21 c4 99 1a d5 a0  8e
10e9 : 9d 26 41 da 59 04 e6 8c  af
10f1 : 92 7f c0 e9 69 25 16 be  46
10f9 : 4a 5e b4 80 e1 6c 26 9b  01
1101 : 07 0b 61 35 b1 3b 4b 45  39
1109 : 12 6a c4 cd 08 d2 aa 4c  95
1111 : d4 69 b4 15 d8 4d a1 d3  90
1119 : d8 07 c5 b0 9a d8 85 9e  c0
1121 : c0 3e 2d 84 d6 08 59 ec  c9
1129 : 26 c7 4b 36 32 56 a2 0c  45
1131 : 0e 24 12 03 73 93 a4 07  ab
1139 : 5b 05 9b 9c 9a d1 24 98  8b
1141 : f8 9a 5f 6e 4b 4f 6e 4b  ab
1149 : 16 8b 05 d6 c2 24 93 1f  1b
1151 : 13 4a b2 cb 30 d4 67 49  89
1159 : d5 bc ef d2 b1 a1 d5 3d  dc
1161 : 9d 8e 55 5a 34 dd ce ac  ad
1169 : 3f 26 28 f9 aa 34 5d 03  cc
1171 : e3 ac 45 02 6f ff 03 a9  92
1179 : a0 cf 24 f9 04 45 4e 49  7f
1181 : e4 7a 32 43 33 e8 65 f1  8b
1189 : e6 6c 4f 08 13 af 9b 53  3e
1191 : 92 7b 2f 8f 17 cc 89 e3  64
1199 : 40 46 a1 fd b2 08 51 d2  7b
11a1 : 87 85 4a 33 21 cf a5 0f  29
11a9 : 0a 94 77 e1 cf a5 3a 4a  bf
11b1 : 4d 0e 99 90 76 fe 41 0a  f6
11b9 : 32 43 c2 a5 1a 40 e7 d2  db
11c1 : 87 85 4a f0 ad e6 43 9f  1a
11c9 : 4a 74 ca b3 e8 8c cc d1  40
11d1 : f1 37 f6 d4 86 d8 07 e0  c3
11d9 : 96 70 d5 69 52 b1 0d bd  ac
11e1 : 59 21 84 c4 43 b0 4d ae  d1
11e9 : 9b 1d fc d8 cc f6 a6 4a  21
11f1 : 1f 82 59 d8 26 b1 0d bd  62
11f9 : 59 36 81 31 10 e1 59 5e  26
1201 : 46 96 7c a3 a5 59 94 55  48
1209 : 22 23 f2 ad c9 21 47 e5  be
1211 : 1f 90 7c d9 53 1f 6c 8f  d2
1219 : 87 04 7f 58 7c d1 c0 e8  b8
1221 : ec 95 59 e7 2a 15 56 4a  64
1229 : ab 9e 3b 11 8d 5a 14 51  b3
1231 : e8 b7 26 74 42 a1 e9 ec  c0
1239 : 3e ed 17 c2 67 44 2a 1f  0b
1241 : 6f 61 4f 55 c9 33 42 22  63
1249 : 3e 69 0a 3f 22 4c 55 7f  7f
1251 : 63 34 22 23 f2 10 a3 e6  c7
1259 : a5 ea c1 61 6a 25 8d b9  89
1261 : a4 1b 86 2e 47 c7 3a e1  5a
1269 : 84 a8 b2 8d f8 3a 81 7d  02
1271 : f9 b9 38 a5 55 db 24 28  1f
1279 : fe b6 4a ab 19 ea 15 53  be
1281 : be 44 9d 05 45 3a 1b 0f  1a
1289 : 78 ef 35 29 e0 e5 c7 0d  e2
1291 : 5c 70 9e 8e 86 ce 9c 8e  0d
1299 : 54 2a b8 a5 55 c9 9d c5  8b
12a1 : e4 87 bc 6c 07 50 b1 f3  a7
12a9 : b4 1d a2 b3 16 23 42 cb  26
12b1 : 45 d2 fc e1 a5 7a 48 d8  dc
12b9 : a5 55 2d 8b 4d 91 55 0f  9a
12c1 : f2 8e 86 96 38 7c d1 e7  ed
12c9 : 43 49 4e c2 26 8b 35 60  f1
12d1 : 75 09 a7 83 49 4e 17 d1  2c
12d9 : e7 66 ab 43 57 1c 34 94  97
12e1 : e1 7d 1d 84 4d 88 89 2c  f0
12e9 : 46 84 2f a3 b0 89 b7 d7  97
12f1 : bb 73 fc 6c dd b9 f8 0d  dc
12f9 : 80 e8 e6 6a 1c b8 e1 f3  eb
1301 : 47 3b d0 31 43 11 17 01  5b
1309 : b1 71 6d 42 6f 50 34 94  8a
1311 : e1 7d 1d 84 52 cb ea 04  c0
1319 : 1e b7 25 0b 98 e8 6c e9  14
1321 : d9 2a ac 70 54 2a ab 32  f2
1329 : 55 67 8f 50 aa ac c9 55  02
1331 : 9e b1 50 aa ac c9 55 62  44
1339 : b5 42 aa b3 25 57 70 55  aa
1341 : 4d 82 b2 00 f4 34 94 ec  99
1349 : 46 34 50 e5 c7 61 13 45  d8
1351 : 22 2a 36 54 34 94 ec ec  16
1359 : 95 63 9c 69 e0 be 8e c4  bc
1361 : 64 b3 cd d7 03 a3 ca 69  58
1369 : 2c dc 10 e5 c7 0d 25 38  ae
1371 : 5f 47 62 32 59 64 28 e0  6e
1379 : b2 55 75 48 ce b0 79 b2  fa
1381 : e9 e9 e9 fd 91 0d 25 38  1f
1389 : 5f 47 62 32 59 76 d4 26  54
1391 : fa 82 89 e9 c2 b7 24 2c  3f
1399 : 36 d8 0d b3 b4 3c 6c 0e  f0
13a1 : 21 d7 0c 25 44 92 4d 50  04
13a9 : d9 d3 85 14 7e a0 7c d4  d8
13b1 : c8 54 53 90 aa 15 54 36  9b
13b9 : 1e f2 1e 65 29 df ad 27  1b
13c1 : 4c 95 57 33 cb 32 55 5c  70
13c9 : ce 21 e6 52 a1 ea 52 ca  7c
13d1 : 2e 2c 2b bf b7 f9 09 63  0e
13d9 : b2 2a e8 53 b8 a1 56 7d  32
13e1 : 13 73 66 4d 82 c9 57 62  8a
13e9 : b5 42 af 42 ac fa 26 e6  fc
13f1 : cc 9b 06 5b 9f dd 95 cd  53
13f9 : 32 49 64 72 7f de 51 50  0c
1401 : 72 f6 95 e1 65 16 92 a4  2a
1409 : b4 a6 9c a4 b7 29 be 6c  65
1411 : b2 a6 6d 8d b9 4d ab ae  35
1419 : 2e 71 4d ae a5 34 78 97  36
1421 : 72 7d a2 14 78 5a e9 40  ff
1429 : ad b7 b5 e9 70 93 36 c6  66
1431 : 59 b2 e7 cb 06 ab da 37  ee
1439 : 85 95 7c 6d 31 11 1f 6a  43
1441 : 25 dc ab 0a 15 79 1e d3  3e
1449 : 7e 12 66 ad 89 ed 4b 36  c1
1451 : 84 ab 0b 6f 6a 66 85 5e  08
1459 : 47 b5 b6 e1 1b 51 aa 7a  40
1461 : df 85 b4 e1 37 e1 1a c6  e5
1469 : 84 d4 f2 cd bb 49 aa 1e  ba
1471 : 95 3c a9 8f b6 c9 57 62  5c
1479 : b5 42 ae 44 54 c2 54 98  61
1481 : ea 6c 95 53 78 ab 25 55  95
1489 : 85 9a 76 3b 1d 89 e9 03  2c
1491 : c9 d5 2c 3f da 51 5e 14  12
1499 : 85 a9 81 4b af ca 71 53  7a
14a1 : 02 95 3b e6 c6 48 85 a9  32
14a9 : 81 4a 43 ea 60 52 92 de  1e
14b1 : a6 05 29 0f a9 80 47 b2  27
14b9 : b5 19 bf 61 f2 dc ce 4a  fd
14c1 : 6d 53 ee 54 ea ea 77 12  26
14c9 : dc d6 ca 6d 27 f6 [illegible] 3f  f9
14d1 : 89 f4 b2 7e e9 1e 8b 23  55
14d9 : fe a4 b8 53 d1 3e 55 cf  c6
14e1 : 73 b2 7d 2c 9d 49 26 46  db
14e9 : b4 99 4a 66 1e 47 aa 43  17
14f1 : 3c 88 8a e3 d1 1c b0 f1  35
14f9 : 7e f0 4e 70 c3 49 a0 4e  37
1501 : fc 79 04 f7 87 d9 23 41  10
1509 : 9f 23 61 f1 89 f1 85 7c  08
1511 : 42 15 42 aa 4f 3d 43 d5  5b
1519 : c5 2a ae fa 3d 46 4a ab  85
1521 : 1b b5 0a ab 38 67 a4 d0  02
1529 : 85 53 0c f1 f8 83 f9 d7  dc
1531 : f5 0e f9 52 6a 87 99 4b  d6
1539 : d4 26 f0 7b 64 28 dd 8e  e8
1541 : d6 17 6e f7 ec 3e 89 ea  fa
1549 : 4e 58 dc 8e 38 72 b4 d0  58
1551 : 6f 7c c3 d8 46 0e 5d dd  10
1559 : b7 6d cf d4 f2 36 bc a5  74
1561 : 72 d9 84 60 e5 dd cc b9  e1
1569 : c4 47 92 75 a0 c9 4e e2  7b
1571 : d0 e5 d2 f7 c3 be eb 1c  01
1579 : bb 9d e1 35 02 e4 88 8f  aa
1581 : 7a 2b 93 00 5b 11 11 e8  ca
1589 : 9d 24 6d 19 2a ec 56 c2  a0
1591 : bb 48 f9 eb 1d 5f 68 f4  c4
1599 : bd 66 4a be 53 e3 35 43  a3
15a1 : fc 7f ce ad b2 ae f5 12  63
15a9 : 9e 77 1b 2b c3 ed 1e 6b  2a
15b1 : 1e 9d d1 f2 9c f7 3b 47  75
15b9 : a8 55 d0 7b ef 10 a3 e3  85
15c1 : 91 10 e0 6d 88 50 4f 22  cd
15c9 : 16 39 0b 9b a2 78 72 b4  d3
15d1 : d0 4e b4 cd f3 0f 22 21  32
15d9 : 59 4c 4b 10 ac a6 22 21  f8
15e1 : 59 4c 15 94 df a3 92 50  3e
15e9 : ac a6 21 42 b2 98 34 9e  77
15f1 : 40 72 b4 cd 20 ee a0 6e  2a
15f9 : a0 10 a1 59 4c 44 42 b2  8a
1601 : 98 2b 29 83 87 6d 35 8a  b7
1609 : cf 6c f0 ac a6 67 e3 3c  8e
1611 : 2b 29 83 49 e4 10 a1 59  e3
1619 : 4c 6c 90 e5 69 9a 41 21  2f
1621 : 6a 60 12 66 f3 1d 4b 33  c8
1629 : e0 68 14 e1 59 9e d9 e1  34
1631 : 59 4c cf c6 78 56 53 06  11
1639 : 93 c8 0e 56 99 a4 1d d4  5c
1641 : 0d d4 02 14 2b 29 82 b2  27
1649 : 98 2b 29 83 95 a6 69 ae  c3
1651 : d7 c0 2c 2c 7c dd 13 ea  f2
1659 : f4 4f 43 65 e4 3c 2e bd  d7
1661 : f9 e1 59 4c 9d f7 aa 6d  4a
1669 : 3a c1 fa 76 c9 22 b0 8d  9d
1671 : e6 3e bf 44 f4 ac 48 fa  ba
1679 : 06 81 4e 15 a4 05 3f fc  df
1681 : cd ac 1b 6a f0 f5 cc 4e  47
1689 : 72 91 4f 0f 9b a0 42 a9  95
1691 : ff e3 c3 ee d5 1b cb 21  f8
1699 : 46 81 91 70 2f 83 77 e4  c9
16a1 : 10 aa 45 3c 1e f4 10 ae  06
16a9 : af 4d fd 74 84 df d7 48  44
16b1 : 79 df 3f ae 49 a0 be 4d  ef
16b9 : fd 70 d0 06 ef c8 9b 1a  cb
16c1 : 09 b1 7c 1b bf 20 23 5b  66
16c9 : 25 ff af 0a fe b9 de 40  d5
16d1 : 7f b2 68 64 79 01 f0 db  6b
16d9 : 84 d6 78 70 b2 08 50 4f  40
16e1 : 20 3f 4e d8 5f 07 e9 db  dd
16e9 : 0d 01 cf 8c d5 0f af aa  e6
16f1 : 13 59 e1 d3 c8 43 a1 e4  9b
16f9 : 06 a3 ff 4e e0 39 6c 4f  c3
1701 : e0 82 75 ec 22 36 80 5a  a8
1709 : f3 ad 1a 41 fb bf dc ff  b3
1711 : bf fa 8f 4b cf d7 94 fa  9f
1719 : 9d 46 bd 13 ea 7b e0 b0  1a
1721 : eb 11 11 7c 16 c9 83 64  ef
1729 : 90 ec f4 03 d7 34 2b 4b  2f
1731 : d4 51 dc 5b b8 49 aa 1d  0b
1739 : bb 70 f9 ba 07 45 d7 34  64
1741 : 82 cc cb 8a 98 1a bf 34  2f
1749 : f2 7a a1 f8 ad 9e 6a 21  bb
1751 : 5a 33 6e e3 35 14 69 1a  2b
1759 : 41 ce d4 37 ae 4f 3b be  ed
1761 : 6f 5c 9a 64 de b7 ae 4d  32
1769 : 9b 77 0d eb 9e 1e de e9  ab
1771 : 91 70 2f 83 95 9e 44 46  62
1779 : 80 39 59 e0 f7 a0 0d 92  e6
1781 : 78 89 1e b7 ba 0f c3 91  93
1789 : 6d 40 0b 64 21 5d 5e 1b  12
1791 : 64 9e 14 9a 86 f7 49 d0  8c
1799 : b7 ba 4d 34 c8 51 bb 1d  c7
17a1 : a8 fa 54 33 b3 a1 cf d0  6b
17a9 : 6f 74 9c 1c 6f 06 ef c1  67
17b1 : da f8 3e 94 b3 65 e7 5d  ea
17b9 : b7 ba 4e 97 29 ac d6 5d  62
17c1 : f4 d4 d6 e9 0e 11 83 95  b5
17c9 : a6 83 7b e6 1e c2 23 69  44
17d1 : 79 69 af da 1e 64 56 21  e7
17d9 : 60 6d 0f 0f 47 3c eb 46  28
17e1 : 92 00 e4 a6 d5 9f 44 0e  09
17e9 : 48 e7 d2 08 88 be 68 d0  9c
17f1 : 6e 80 34 22 3e 39 f6 e3  42
17f9 : d4 71 34 94 d8 be 42 92  d7
1801 : dc 2e 6b 20 86 99 28 51  4c
1809 : 6f 72 9c 0f e2 21 46 e9  de
1811 : 68 56 97 b3 87 a9 db 3e  b2
1819 : f7 9a 7a 9f 19 eb 05 66  c2
1821 : 78 53 67 e3 1b bf 7f 75  32
1829 : d1 8f 9e de e8 26 73 ef  b3
1831 : 2e d5 7f 75 3e 8b 7b a3  4e
1839 : 9f 6e 05 c2 5f c5 c4 e1  a4
1841 : e6 6a e3 9a 7a d0 ad 08  9d
1849 : 8e 6e d0 f5 82 b3 3c 29  0a
1851 : b3 d2 d6 3f ba fe 91 f3  dd
1859 : db dd 04 ce 7d e5 db fb  6c
1861 : ac 68 e8 b7 ba 39 f6 e0  85
1869 : 5c 25 c4 bd 9c 3c cd 6a  f8
1871 : f9 a7 d5 ce 1f 7b cd 3c  0a
1879 : d8 57 b3 87 d1 35 c0 6c  7d
1881 : 4e d8 7b b1 16 0e f5 2a  4e
1889 : 11 bb 3a 82 76 76 77 1b  86
1891 : 91 77 ae 3a 81 8b 08 2f  c4
1899 : ee bc b8 f9 eb 22 7c 6b  eb
18a1 : 17 21 ce 7c 47 b8 b1 4f  2c
18a9 : 4b 09 cb 4d ad 89 0a a4  ae
18b1 : 53 d0 a4 b7 58 3d 1c f7  5c
18b9 : 85 a1 6f 74 99 b6 2b 1d  af
18c1 : 77 20 ed ac 88 56 99 cf  9a
18c9 : 52 6c 4f d4 12 22 05 cd  a2
18d1 : 60 dd d2 34 1d ac 13 9f  1e
18d9 : 64 e6 9e 97 12 96 4e d7  0a
18e1 : 27 34 f4 9a 87 98 91 db  ee
18e9 : 58 56 99 bf b0 f6 8d 15  ee
18f1 : 9f f6 c6 e8 87 cf 94 7f  22
18f9 : 13 c2 cd 8e d6 6c 6e d6  eb
1901 : 85 76 1a 64 8e b4 82 d3  15
1909 : a3 fb 0f d5 22 c3 51 da  64
1911 : 81 c9 1d 15 a4 13 9f 34  2b
1919 : 69 53 00 ed ab 46 9a ef  21
1921 : 6e 04 0f 6c f2 16 82 06  d9
1929 : 5c 04 e8 58 6a 6c f8 5b  71
1931 : 53 9f 66 e6 9e cc 53 cc  01
1939 : d5 0a 4b 79 e7 ca 45 3d  7a
1941 : 6d 76 bf 34 f4 bc 09 04  c1
1949 : 2c 0c d3 ce 8a d2 0c db  71
```

Listing 2. »DOS V3« – das neue Betriebssystem in der Floppy. Bitte mit dem MSE eingeben

```
1951 : 46 95 30 28 5a 34 d7 6b  f0
1959 : fd b3 e8 20 10 b0 3d b3  51
1961 : f2 e0 27 42 c3 53 74 16  aa
1969 : 3a d1 a4 12 17 37 34 fa  e9
1971 : 5c dc d3 d6 ce f7 28 9e  95
1979 : 9b c0 a4 90 1d ce ac 17  d9
1981 : 2d 89 b5 04 eb e4 c5 9b  95
1989 : d7 52 4f 3e 81 ee 15 a4  52
1991 : 16 45 c5 4b a2 16 a6 c1  1e
1999 : 09 36 ec 5c 8c 55 88 ea  ef
19a1 : 15 a1 9a ab 3f 60 b0 5c  15
19a9 : d6 41 58 36 f5 64 6a 37  97
19b1 : 64 47 65 14 f5 22 9e 75  6b
19b9 : dc 80 ef 52 9b 1b b7 f7  7d
19c1 : 46 f0 8e 7d b8 17 09 75  26
19c9 : e6 97 88 f8 e7 db 8f 51  fa
19d1 : c1 b9 9a b1 71 2e 4b e2  c7
19d9 : 14 68 2d ae cd d6 27 6c  4b
19e1 : 13 af b6 7c 69 7a b1 71  1d
19e9 : 4b 25 f1 0a 34 16 d7 66  a5
19f1 : ff b4 b6 07 50 2b c8 9f  9a
19f9 : 1a 03 9f 07 3f 41 4d 99  c4
1a01 : ea 74 cf aa a0 c0 c2 c6  17
1a09 : 3a 83 c8 d9 9e 34 09 1f  60
1a11 : 6d 40 13 95 89 35 42 6b  38
1a19 : 3c 85 1a 00 9c ac 37 78  9b
1a21 : 86 9e b0 88 ec a2 9e a4  db
1a29 : 53 ce 93 20 88 8c 96 45  9e
1a31 : c6 36 ee 48 be 9a 9e 4a  a7
1a39 : c0 3b d4 9f dd 1b d5 64  97
1a41 : 5c 2b c8 92 2f a2 49 dc  9e
1a49 : 7a 8e 26 9b 13 e2 09 88  85
1a51 : 81 de a4 fe e9 2e ac d8  bf
1a59 : c9 21 46 80 27 2b 12 6a  3d
1a61 : 84 d6 79 3d 20 98 88 1d  79
1a69 : ea 54 32 00 34 0e 7c 1c  e8
1a71 : fd 05 36 67 cd 8d d9 d4  c6
1a79 : 1b 8d 99 f8 13 c1 6f 9e  1a
1a81 : 67 51 28 0c 91 0a b1 d7  fc
1a89 : 72 08 51 bb fb b2 5c 4c  2a
1a91 : ca b0 7f 75 e5 c7 cf 48  ae
1a99 : f8 97 72 dc f7 93 26 cf  e9
1aa1 : 7f 75 55 76 68 d3 63 76  9f
1aa9 : b1 3d 20 98 89 ab dd 38  f2
1ab1 : c9 dd 38 97 81 39 7b b6  a7
1ab9 : 70 b2 c4 bb 50 f5 d8 ea  19
1ac1 : cc 72 14 2a ec e1 2e c3  2f
1ac9 : 89 ee 82 98 4b 93 8c a7  50
1ad1 : 11 72 15 47 4b 93 c0 52  43
1ad9 : c1 75 1b 2e 4b 45 37 8b  b4
1ae1 : 92 f9 4d 92 ea 6f 17 23  e2
1ae9 : d1 db 8e d0 96 23 e1 12  94
1af1 : c0 7e d1 1b 0f 88 49 c1  a6
1af9 : f1 48 d4 7f 09 11 1f 0c  e1
1b01 : 8e 47 e1 fc 92 75 4c c6  de
1b09 : 5d 9c 25 e7 b7 8a 61 2e  ac
1b11 : f1 b6 bc a5 73 51 44 63  db
1b19 : 46 a1 43 6b 88 8b 08 a5  be
1b21 : 61 71 b9 25 ca 13 f1 60  1c
1b29 : 8f 97 55 b7 49 a0 7f f6  56
1b31 : 29 de 58 ed d0 e5 c5 38  61
1b39 : 1d b5 68 e6 79 0e 5c 59  54
1b41 : 97 32 d0 e5 c5 20 08 3c  d8
1b49 : 6d 71 b9 0e 56 22 2e fc  c8
1b51 : fb 6d cb 9a db ce 30 54  e7
1b59 : e2 22 77 ae 33 ae a7 48  d8
1b61 : b0 83 fd e8 f7 e8 72 e2  c5
1b69 : b0 8d 57 96 6c a3 bb f4  45
1b71 : b5 18 d5 9b d4 b6 5a a7  d7
1b79 : a3 eb c4 f5 ce dc 8f 19  46
1b81 : 07 58 88 ca 8c ff 7a 3c  5b
1b89 : ec 9d d3 89 78 13 97 bb  60
1b91 : 67 03 ad a4 c4 6c d0 74  56
1b99 : 59 2a ec 7f 93 6d 15 0a  3f
1ba1 : bc 2f 2c e9 8a 2e 72 5e  de
1ba9 : fd 12 c9 35 41 fe 22 27  ab
1bb1 : 6e 55 9f 6a 37 e4 28 bb  b2
1bb9 : ed 3b 29 16 10 64 ab b1  87
1bc1 : 99 67 a2 14 1b 01 9d 22  ad
1bc9 : 23 df a1 3f 16 f3 73 af  5a
1bd1 : 29 64 39 72 c7 24 66 28  50
1bd9 : 5a 4f ba 1d ed 0a b3 ed  07
1be1 : 46 fd e1 6e 1d cd ad 2b  b9
1be9 : d8 93 cf 94 52 51 8f 14  27
1bf1 : 3b 3a 78 6e 2d 59 45 7a  dd
1bf9 : ae 29 b7 30 dc 5a b2 8a  b0
1c01 : f5 3e 05 2e 55 81 b1 11  67
1c09 : 0f 4f bc 79 18 85 d2 79  0a
1c11 : 25 99 73 39 44 53 63 42  f8
1c19 : db 9c ad 4f 7e 74 c5 13  60
1c21 : b4 b5 27 6d da 6b b5 fa  fd
1c29 : 82 d5 79 26 d4 eb fb e2  1b
1c31 : d5 79 c6 f3 90 f4 3b cb  28
1c39 : 6e 79 b0 08 78 4f c5 3b  21
1c41 : c9 1d 41 de 70 f7 c4 47  2d
1c49 : 14 9c 4c 52 4f 21 3f 14  2c
1c51 : 96 82 81 dd 75 67 cf aa  6b
1c59 : d6 cd 1f bb d6 8f 4e 53  1f
1c61 : 69 3b 99 4f 43 95 88 8b  d2
1c69 : 9d 11 d1 61 1a ad b9 bf  a5
1c71 : 93 c2 bf de d5 73 8d be  de
1c79 : 3a 62 89 64 b0 9e 23 8c  79
1c81 : ff 7b bb e9 b9 6a af 23  5e
1c89 : 5b 54 86 06 a6 41 d3 09  47
1c91 : 7e 79 ba 29 e9 af cf 45  85
1c99 : 85 80 e4 50 3b 73 06 7d  04
1ca1 : a8 df 99 b4 dd 13 a9 a4  9c
1ca9 : ad b7 2e 7f ea 91 cb 54  c0
1cb1 : 39 71 64 ee 43 95 88 89  b0
1cb9 : d3 08 4b ef 0e 77 20 eb  56
1cc1 : f9 04 e2 8b b2 14 6f c8  01
1cc9 : 51 df 04 4a d3 b0 c5 53  d5
1cd1 : ac c7 ae 75 90 f9 e1 12  40
1cd9 : b4 ea 31 f3 8e b2 1f 26  14
1ce1 : 6c 6f e6 c5 da d4 45 49  73
1ce9 : aa 1f 8b 4e 0d 34 f0 dc  bf
1cf1 : 5a b2 8a ed dd a6 6d 3b  44
1cf9 : cf c9 0d c5 ab 28 af 00  64
1d01 : a8 a6 6e f0 32 b0 4e c6  25
1d09 : aa fb 49 36 24 64 27 e2  92
1d11 : 84 7d 3d 57 a2 93 52 36  0b
1d19 : c5 22 28 4f c5 07 3a 74  4a
1d21 : 3b 3a 73 bc b6 90 73 b5  17
1d29 : 03 e4 e7 77 c3 e7 3c 2b  4a
1d31 : fd e8 aa 9b fd e8 f5 c8  51
1d39 : 59 8f 26 63 ce 98 56 63  22
1d41 : ae 3e 78 6f a2 ca f6 03  7d
1d49 : b3 a7 78 58 6e 2d 59 45  39
1d51 : 7f a7 8a 53 3a 4e fa c4  3c
1d59 : c9 2d 2c db 02 1e 84 fc  5c
1d61 : 52 19 46 ca 91 ae 2b 65  31
1d69 : 5c d6 62 95 77 79 6c a2  b6
1d71 : bd ab c2 ad 44 e5 26 a8  c7
1d79 : 37 cb 08 fa 71 f4 59 2a  70
1d81 : ec 39 33 62 58 a1 3f 14  db
1d89 : 3f 1d 68 39 d3 ca 9e 86  b3
1d91 : e2 d6 59 a7 08 f5 46 ab  ca
1d99 : b3 84 bb 1f 04 f5 f2 98  4e
1da1 : 4b 91 ec a6 ca 22 bd ac  d3
1da9 : 37 16 b2 cd 38 f3 bb af  c3
1db1 : 2a 30 dc 5a a8 8a e9 9b  33
1db9 : bc 7e ee 1b 8b 55 11 5d  36
1dc1 : 33 69 e1 b8 67 09 79 ec  b7
1dc9 : a5 30 97 21 54 6a b8 68  dd
1dd1 : 69 ef 37 3a f2 97 54 ee  62
1dd9 : 01 4a 83 fc 44 48 5d 76  69
1de1 : d1 da 66 d8 ac 42 8d 40  68
1de9 : bb 1c c3 af c7 66 [illegible] [illegible]  e6
1df1 : fc 7a 44 ee 8e f8 75 81  a3
1df9 : 12 b4 ea 31 c6 bc d6 1d  2e
1e01 : 64 de a1 33 ac db d4 26  ec
1e09 : 76 1b 7a 84 ee 35 81 12  ff
1e11 : b6 f3 58 75 1b 7a 84 ce  bb
1e19 : b2 6f 50 9a 5b c3 ac db  28
1e21 : d4 26 76 1b 7a 84 ee 35  fb
1e29 : 86 e6 1d 66 dd da 67 61  4b
1e31 : b7 76 9d c6 b0 22 56 de  97
1e39 : 6b 0e a3 6e ed 33 ac 9b  c4
1e41 : bb 4c eb 36 ee d3 3b 0d  78
1e49 : bb b4 ee 35 97 dd d9 ae  ee
1e51 : 61 26 06 a0 eb f1 e9 3c  c9
1e59 : fc 8f dd b7 a8 99 34 57  62
1e61 : ee db d4 4a ee de 95 16  24
1e69 : ee d3 a5 35 6c 98 6b 51  2d
1e71 : 02 4d 50 97 96 11 f4 e3  ae
1e79 : e8 87 e3 ad 06 9a 77 9d  22
1e81 : dd 79 51 86 e2 d7 a6 8a  dd
1e89 : e6 f7 d3 36 83 f7 70 dc  9a
1e91 : 5a f4 d1 5c d6 29 df 70  7c
1e99 : 4d 58 82 bd e1 2c 10 9f  6a
1ea1 : 8a 0e 74 e8 2b de 10 a3  9e
1ea9 : c7 1f 62 f3 7c 75 1b 09  09
1eb1 : 62 46 9d 83 7b 69 5a d1  1e
1eb9 : a3 41 ec 9d 46 c5 cc 1b  e8
1ec1 : db 4a df 1b d9 e3 f8 de  7b
1ec9 : cf 1d 8b 06 dd 96 ee d3  c0
1ed1 : 3a c9 84 b1 23 4e c1 bd  6e
1ed9 : b4 ad 68 db 8e 3c 73 ac  eb
1ee1 : 98 b9 83 7b 69 5b 89 1b  74
1ee9 : b0 3d 4e c3 08 b6 e3 b7  79
1ef1 : 76 99 d6 6c 25 89 1a 76  6b
1ef9 : 0d ed a5 6a be 33 6e a3  3a
1f01 : 77 69 f8 cc 3e 1d 21 d6  03
1f09 : 6c 5c c1 bd b4 ad 68 d1  c9
1f11 : 8f 95 b6 90 74 87 61 b0  95
1f19 : 96 24 69 d8 37 b6 95 be  34
1f21 : 37 b3 ec 78 de cf b1 e3  77
1f29 : 7b 3e c3 0d 5b 69 1b bb  3b
1f31 : 4c d1 8e e9 0d f9 d8 6c  23
1f39 : 5c c1 bd b4 ad 6d ec 37  e4
1f41 : 76 9d c6 f9 66 9c 4d 47  86
1f49 : 7d bc b9 f2 7c e9 ee bf  43
1f51 : 9f ea ec ba 3a 9f 7c 85  95
1f59 : 17 63 50 3a c2 77 47 7c  7b
1f61 : 30 ce bf 1e 91 11 1b f3  a2
1f69 : be d0 d1 de 6b 1f d4 0d  fd
1f71 : 92 6f ea 06 cd 37 f5 03  ab
1f79 : 86 9d c6 b0 0d 97 79 ac  62
1f81 : 7f 50 34 69 a5 de bf a8  04
1f89 : 1b 24 df d4 0d 9a 6f ea  82
1f91 : 07 0d 33 73 0e f4 46 c8  8d
1f99 : 85 1b 31 c3 40 d9 77 9b  58
1fa1 : e7 ee e7 67 67 46 8d fb  bd
1fa9 : b6 2c 1a 41 4c ce 8e d9  4d
1fb1 : 8d fd e6 63 f7 6d 26 91  09
1fb9 : a5 69 49 d8 83 7b 7d 83  91
1fc1 : 0a 8f a2 f3 04 68 df bb  34
1fc9 : 63 73 3a 36 49 bf 77 e3  73
1fd1 : 4e ce ce c1 bd be c1 85  56
1fd9 : 65 d1 79 82 36 4f dd b1  8e
1fe1 : 72 d2 0a 6f dd b9 46 95  1d
1fe9 : 8f 96 76 74 6c ef 5e 60  70
1ff1 : c2 b3 e8 df 60 8d 9b f7  94
1ff9 : 6c 35 1c 34 cc e8 ed 98  8a
2001 : df de 66 3f 76 d1 8b e6  c3
2009 : d8 7d 8e c7 62 0b f7 6c  74
2011 : 7c 09 dc 6f b5 d7 af 30  70
2019 : 7a f7 d8 23 87 79 be 5b  1f
2021 : 11 b4 e1 da da a4 0c d5  0f
2029 : 0d 67 34 37 8e 25 17 b9  c0
2031 : c5 75 29 b7 73 c0 74 47  90
2039 : aa 22 bc 37 16 b2 cd 38  a9
2041 : 05 70 85 0e c7 12 79 eb  6c
2049 : e2 b9 7b 7d ce 86 f5 e8  61
2051 : 3b 7a 92 db 57 b3 4f 30  9a
2059 : 1e 9b 2b 11 a3 db 12 a2  d8
2061 : f7 0a 33 67 af 4f 6a f6  24
2069 : e5 d5 83 d9 e6 9c 47 bb  3d
2071 : 2d e3 d2 14 77 c3 50 3a  12
2079 : c2 22 3b 32 7e 41 cc 3a  fb
2081 : fc 7a 41 12 b4 ea 31 c6  42
2089 : bc d6 1d 64 de a1 33 ac  a5
2091 : db d4 26 76 1b 7a 84 ee  a4
2099 : 35 81 12 b6 f3 58 75 1b  f8
20a1 : 7a 84 d2 ef 4e b2 6f 50  e9
20a9 : 99 d6 6d ea 13 3b 0d bd  21
20b1 : 42 66 e6 1d 64 3e 89 dd  9e
20b9 : 21 bf 58 eb f1 e9 10 a3  43
20c1 : 7e 4e 28 ef 8e f0 86 8c  12
20c9 : ef 3c 6c 8e eb 8d 99 ca  ea
20d1 : 8e 18 1b 2c ef 38 68 ce  b8
20d9 : f2 46 c8 85 1b 31 c3 03  21
20e1 : 65 9d d2 1b f5 af 37 30  47
20e9 : 0a 53 a1 a4 d4 34 34 e1  1d
20f1 : f8 e2 c2 d5 7d b4 3e 7d  37
20f9 : 02 3c 10 64 ab cf 46 50  9d
2101 : ab 93 c9 19 0b 42 db 9c  77
2109 : 13 ce 44 08 f4 64 2c cd  54
2111 : 82 ea 36 5d 92 ae c3 9a  24
2119 : ad 48 ea 52 8c 4a 01 5d  c9
2121 : 97 07 a5 ef d1 1e 6b 02  63
2129 : 15 47 07 a4 d0 67 48 7a  97
2131 : 8e 0f 42 a2 72 16 dd dd  37
2139 : 81 c1 ae a2 03 35 43 e7  51
2141 : 9c b6 90 6e 8d 0b 98 cc  57
2149 : 5a 08 30 e5 2d 43 25 5d  ac
2151 : 8d 8a 85 5c 8f 47 6e 42  82
2159 : 88 25 fb e8 58 e7 19 aa  0e
2161 : 1f 3c e7 6b c4 29 d3 ae  48
2169 : 5b 6a 9e ab 3e 8d 57 e1  88
2171 : c0 68 4f e2 3e 21 27 07  2d
2179 : c5 36 a3 77 e1 ee fc 31  1d
2181 : fe a7 78 62 b1 3f b8 f8  a7
2189 : 84 9c 1f 14 e9 d7 37 f1  c4
2191 : 27 75 65 5e 5a 13 f3 87  b5
2199 : 4e 9b 3c 9b 31 f1 09 38  6f
21a1 : 3e 29 0f 15 90 f9 e7 2d  ad
21a9 : 96 37 68 e0 63 f2 88 11  23
21b1 : b0 f8 84 9c 1f 15 0f 9e  a6
21b9 : 72 da 77 56 55 e9 69 18  bb
21c1 : 6a 3c f6 4a bb 1c d6 dc  02
21c9 : f5 0a ba 1f 3c e6 33 36  8a
21d1 : 0b ee fe 15 1b 2f 6c b1  f6
21d9 : b0 8e f0 c5 62 7a 63 e2  12
21e1 : 12 70 7c 54 2a 27 2d bf  e5
21e9 : a9 a2 90 e1 f1 09 38 3e  08
21f1 : 29 b3 d9 b0 5d 5e 47 c2  ec
21f9 : a3 65 ed 96 37 00 9b 31  e1
2201 : f1 09 38 3e 29 0f 15 99  df
2209 : c2 5d 8e 0a 98 4b ac 76  e2
2211 : d1 cf e7 22 c2 0c 95 70  cc
2219 : a0 d8 0c e9 11 1e fc 76  48
2221 : 86 68 f0 b7 0f 76 db d4  cc
2229 : 9a 37 de 91 a2 80 e9 0e  3b
2231 : 5c 56 4a bb 0e 4c d8 96  96
2239 : 28 7e ee 0a 1f 26 09 5f  a3
2241 : 66 68 f3 af a5 c9 a1 25  48
2249 : 8a 85 54 29 1e 90 a1 bb  34
2251 : dc 06 de af e7 80 59 72  ab
2259 : 5c c4 5c ea 0c 52 66 0f  97
2261 : 7a 73 e4 b9 f4 5c 5c 0d  c3
2269 : 90 b4 85 40 1c fd 0a aa  ec
2271 : 04 8f c0 74 1c 2d 8a 1d  8b
2279 : 3b 72 65 09 e4 e0 0f ad  55
2281 : 28 f5 3f a1 ea 50 9e 10  74
2289 : 2d e8 8f 9a 96 4b ea af  30
2291 : 59 2b 91 3c 7d 63 8f 77  8c
```

```
2299 : cd 67 fa 1e 27 9b f7 87  da
22a1 : 0b 62 86 e7 74 47 a8 52  45
22a9 : 3d 21 94 6c ae 2b 65 5c  bc
22b1 : f8 38 a5 5d de 5b 28 af  a3
22b9 : 6a f0 ab 15 f6 66 8f 3a  7e
22c1 : fa 9e 9d 92 ae e7 51 0e  4f
22c9 : 2d af 43 c0 74 6c 44 7a  67
22d1 : 1c 57 4e 7a 04 6a 87 ee  0b
22d9 : e0 9d ad 14 10 ca d8 ca  c6
22e1 : 2b c2 da b2 55 dc ef 21  b8
22e9 : c5 b5 a5 36 9d 33 68 11  f0
22f1 : aa 1d 6e 0d 23 45 3c 07  c2
22f9 : 54 3d 85 38 ae 89 f9 7f  72
2301 : 47 11 55 38 8a 93 c5 53  30
2309 : c1 75 0b 5a f0 b4 2a dd  ac
2311 : 78 2e a6 91 4d 39 66 d3  5c
2319 : ea 9e e3 4f 95 5b af 01  2a
2321 : d6 03 5e 1f 15 d4 1a 1d  8f
2329 : 53 dc 69 f2 ab 75 c5 74  09
2331 : e5 b1 23 21 d6 e0 eb a9  53
2339 : 25 17 1b 77 ad c0 6b 32  93
2341 : ed ad 29 b4 fd cf 05 d6  06
2349 : 54 6c 06 c4 6c 87 5b 82  63
2351 : af 23 5a a7 d6 e8 85 53  8f
2359 : ca 74 f0 b0 bc b3 bc 28  5c
2361 : 05 6a 9a 6c f9 4e a0 15  8e
2369 : aa 67 6d 20 15 aa 79 dd  6e
2371 : f4 02 b5 4f 3b 51 00 ad  57
2379 : 53 25 cc 02 b5 4e 01 5a  59
2381 : a6 42 e9 27 0f d2 6c 4f  7f
2389 : d2 73 4f d2 70 4f 81 14  f3
2391 : fe 87 94 e9 9f 29 d5 56  fc
2399 : b6 6b bb 43 22 96 93 34  ea
23a1 : ad a9 4d 38 46 a3 7e 0e  15
23a9 : dc 4d eb a6 43 77 c1 0d  0d
23b1 : 3f 04 8d 86 fc 23 e9 c7  47
23b9 : d1 0f c7 5b 28 b2 77 a0  a6
23c1 : e5 c4 a7 44 7a a2 2b c3  6c
23c9 : 71 6b 2c d3 92 9a 5e 6f  cc
23d1 : 75 e5 6a a2 2b c3 71 6a  93
23d9 : 9c b3 62 23 2a 78 98 6a  e9
23e1 : 22 bc 37 16 b2 cd 39 29  c3
23e9 : b1 1e a8 8a f0 dc 5a cb  1c
23f1 : 34 e3 ce ee bc a8 a8 8a  71
23f9 : f0 dc 5a a6 6e f1 fb b5  94
```

```
2401 : 11 5e 1b 8b 54 cd a7 26  18
2409 : 1a 88 af c3 f2 9d 25 df  3c
2411 : 6a fc 17 50 dc 5a cb 34  01
2419 : e1 db d5 67 da 0e f5 07  4e
2421 : fc d8 6e 2d 61 b8 02 b8  20
2429 : 22 2a 7a 0e 88 51 76 46  3a
2431 : a3 7e 76 b5 4f 29 d0 79  dc
2439 : bb 19 4d 1e 16 1b 8b 56  ad
2441 : 51 5f 6c ec db c4 e5 9b  ad
2449 : 41 53 9d cf b4 17 68 78  2c
2451 : 9c 1e 1f a0 e8 d4 c4 74  09
2459 : 87 5b 82 1e 6e c5 e7 76  94
2461 : 1b 8b 56 51 5f e9 2a 26  3c
2469 : 88 99 b4 25 73 61 b8 b5  20
2471 : 65 15 fe 92 b6 36 e9 39  aa
2479 : 66 ac e1 f9 4e 8d ab 89  00
2481 : b4 3b 69 45 c5 c0 6a 87  f1
2489 : 2e 29 1e 8e d0 85 18 64  08
2491 : 45 09 f8 ac e1 2e c7 c1  61
2499 : 3d 7c a6 12 e4 7a a3 55  5c
24a1 : d4 45 79 40 f0 b0 dc 5a  3b
24a9 : cc 9a 72 59 ab 56 d5 fa  45
24b1 : ce be ef 7e ea 1b 8b 59  12
24b9 : 66 9c 99 73 5a b7 0c 1d  10
24c1 : 24 d5 0e 5c 57 8f 6e 36  f7
24c9 : 89 dc 43 42 7e 2e bc ad  81
24d1 : 60 4c 35 99 35 6b 25 6a  f0
24d9 : a4 c3 55 9e 68 eb fb a8  af
24e1 : b7 77 32 e7 05 66 70 97  52
24e9 : 9e ca 53 09 72 15 46 ab  23
24f1 : ad 2b e0 67 09 76 3e 09  a8
24f9 : eb e5 30 97 23 d5 1a ae  7c
2501 : 4c 35 11 5d e1 61 b8 b5  4f
2509 : 99 34 e4 b7 35 ab 10 a2  23
2511 : ec 6a 07 58 4e e8 ef 8e  08
2519 : bf 1e 91 11 1b fb cd 63  fd
2521 : fa 81 a3 4d fd 40 d9 26  04
2529 : fe a0 6c d3 7f 50 38 69  3b
2531 : a5 63 71 ac 03 65 a1 a2  a1
2539 : e1 03 65 c8 a4 91 43 21  34
2541 : 9d 92 ac 1e 93 50 b2 a5  e8
2549 : 67 64 bb 47 2d 43 03 42  38
2551 : 6f 5d 37 06 6f 9d a8 8d  9f
2559 : 1a 6b b3 aa ea 74 fd 7c  ae
2561 : 38 7e 0e b6 8e cb 67 70  f8
```

```
2569 : da bf 07 d5 ff e5 87 f9  e1
2571 : a5 33 e8 6c bc 83 27 2f  5a
2579 : ef fd ee 06 c1 39 69 f4  59
2581 : 36 99 2b fd c5 e8 f8 fc  90
2589 : 6e 9e 8d 94 3c 34 34 6c  4b
2591 : a1 e1 d1 cc 4c 51 d9 4c  80
2599 : 21 b4 fb 54 68 58 94 bb  b1
25a1 : 9b 4f ca f7 bd d4 77 de  b3
25a9 : ff 1a 37 17 e0 eb ea 11  a1
25b1 : 7a f5 12 b5 8c d9 d5 85  5b
25b9 : 86 f3 b5 c8 2a 85 5d 43  8a
25c1 : d7 43 f5 d3 d2 4a 6d 2c  bf
25c9 : dc 18 2b 06 ae 3a 1c b8  dc
25d1 : ec 9e 7d 17 e8 49 b6 34  6b
25d9 : 7f b0 df 98 7d 05 bb 8d  c5
25e1 : 55 21 ea 15 52 79 ea 1e  fd
25e9 : aa 1e e5 3e 29 55 77 e6  cc
25f1 : 32 d1 74 bb 24 f3 54 3d  4e
25f9 : 56 4a ab 19 aa 15 52 46  ab
2601 : c5 2a a6 d8 d3 d3 d3 ea  a1
2609 : ca ad e9 76 33 b3 b4 fa  8d
2611 : b2 ab d1 2e c6 76 77 56  7e
2619 : 55 7a 25 d8 ce ce d3 ea  98
2621 : ca af 44 bb 11 12 1f b2  cf
2629 : 2a aa 15 56 23 1a 30 e0  3e
2631 : 92 36 29 55 77 74 4f 3d  a6
2639 : 43 d5 38 56 25 88 d0 a1  5d
2641 : f7 e3 de ed cf 11 64 aa  0b
2649 : b1 5a a1 55 38 71 23 4e  73
2651 : c1 22 38 a5 55 df 18 d1  3e
2659 : 7b 85 56 91 a3 17 2a 15  24
2661 : 56 44 ac 5c a8 55 5a 31  91
2669 : 73 4e a1 55 25 8c 2a 13  9c
2671 : 1f 3c 7a 85 55 6c 52 aa  55
2679 : e7 50 77 a0 db 04 c5 94  98
2681 : eb b8 d1 72 3e 96 62 8e  ca
2689 : 0a 07 47 16 c8 79 94 bf  d6
2691 : 8e 2f c3 b8 f4 f7 05 6c  bb
2699 : d5 b3 a6 d3 cd b0 9b 5c  f5
26a1 : b6 45 eb ca 7f 89 c1 40  1a
```

Listing 2. »DOS V3« – das neue Betriebssystem in der Floppy. (Schluß)

Hüllenzauber

Information und Schutz - damit könnte man den Zweck dieses Programms beschreiben. Fertigen Sie sich Ihre eigenen Diskettenhüllen, auf denen sogar der Inhalt der Diskette aufgedruckt ist.

Das Programm »DITA« (Listing) druckt Diskettentaschen und Etiketten. Die Information für beides wird direkt von der Diskette geholt, für welche Tasche und Etikett bestimmt sind. Das Programm ist menügesteuert und wird nur über die <SPACE>-Taste zum Anwählen und <RETURN> zum Ausführen einer Funktion bedient. Das Menü ist so gesteuert, daß die vermutlich als nächste verlangte Funktion vorgeschlagen wird. Zur Bestätigung braucht deshalb im allgemeinen nur die <RETURN>-Taste betätigt zu werden. Sobald eine Diskette beidseitig erfaßt ist, kann die Diskettentasche gedruckt werden. Die Information für die Etiketten wird gespeichert und kann bei Arbeitsende auf Diskette gesichert oder sofort gedruckt werden. Während des Erfassens einer Diskette wird auf dem Monitor das Etikett dargestellt, wie es später auch auf Diskettentasche und Etikett erscheint. Es sind darin der Diskettenname und die ID der jeweiligen Diskettenseite und eine Kennung enthalten. Die Kennung kann über das Menü geändert werden.

Anschließend werden auf dem Bildschirm die Dateien der Diskette angezeigt und können mit <RETURN> (für den Druck) übernommen oder aber mit der <SPACE>-Taste übersprungen werden. Wird bei der Abfrage die Funktionstaste <F7> gedrückt, dann werden die restlichen Dateien dieser Diskette für den Ausdruck auf der Diskettentasche übernommen. Aus Platzgründen sind höchstens 32 Dateien einer Diskettenseite für den Ausdruck auf der Diskettentasche erlaubt. In der Regel dürfte dies ausreichend sein. Es wurde bewußt auf Schmaldruck, Hochstellung und verringerten Zeilenabstand verzichtet, damit das Programm mit praktisch jedem Drucker betrieben werden kann.

Disketten- und Dateinamen werden mit Punkten immer auf 16 Zeichen aufgefüllt. Von dem auf dem Monitor ausgegebenen Dateityp wird nur der erste Buchstabe mit der Kennung, ob es sich um eine geschützte oder eine nicht geschlossene Datei handelt, auf die Diskettentasche gedruckt. Bei Programm-Dateien wird die Ladeadresse ausgegeben. Aus der Ladeadresse geht hervor, ob ein Programm als Basicprogramm geladen und gestartet werden kann (Ladeadresse 2049), oder ob es absolut zu laden ist.

Der Diskettenname, die ID und die Dateinamen werden nur als Zeichen mit den ASCII-Werten zwischen 32 und 128 ausgedruckt. Alle anderen Zeichen werden als Fragezeichen ausgegeben. Beendet wird das Programm mit der Funktionstaste <F8>. Zuvor sollten allerdings die Etiketten gedruckt oder auf Diskette gesichert werden.

»DITA« ist komplett in Basic geschrieben, daher empfiehlt es sich, das Programm zu compilieren.

Nach dem Start des Programms ist das Wort »ERFASSEN« in der ersten Zeile des Bildschirms hell unterlegt. Dieses Hervorheben eines Menüpunktes zeigt an, daß das Programm auf eine Eingabe wartet. Während der Ausführung sind alle Menüpunkte dunkel. Danach ist zu entscheiden, ob eine Diskettendatei aus einem früheren Lauf eingelesen werden soll und ob die Kennung zu korrigieren ist.

Korrektur:

Wird die <SPACE>-Taste zweimal betätigt, dann ist das Wort »KORREKTUR« hell unterlegt. Wird jetzt <RETURN> gedrückt, dann ist zusätzlich unten auf dem Bildschirm das Wort »KENNUNG« hell unterlegt. Nach Drücken von <RETURN> kann die Kennung korrigiert werden. Zugelassen sind alle Tasten, sofern sie keine Commodore-spezifischen Sonderzeichen erzeugen. Erscheint auf dem Monitor ein Fragezeichen, dann wurde die Eingabe zurückgewiesen. Die Taste <CLR/HOME> löscht die gesamte Kennung, die Taste <DEL> jeweils ein Zeichen. Mit <RETURN> ist die Korrektur abgeschlossen. Mit einem weiteren <RETURN> gelangt man in den Modus »ERFASSEN«.

Erfassen:

Nach <RETURN> ist in der unteren Menüzeile »SEITE A« hell unterlegt. Jetzt sollte eine Diskette im eingeschalteten Laufwerk liegen. Nach <RETURN> wird die Diskettenseite erfaßt. Zuerst erscheint im Etikettenfeld der Diskettenname und die ID der Diskettenseite. Danach wird jede Datei revers auf dem Monitor dargestellt. Mit <RETURN> wird die Datei für den Ausdruck übernommen, mit <SPACE> wird der Ausdruck verhindert. Durch Drücken von <F7> wird der Rest der Diskettenseite ohne weitere Anfragen übernommen. Sind alle Dateien der Diskettenseite erfaßt oder schon 32 Dateien übernommen, dann ist jetzt »SEITE B« hell unterlegt. Wurde die Diskette gewendet, wird nun die zweite Seite der Diskette erfaßt. Sobald dieser Vorgang beendet ist, wird durch das Hervorheben von »DRUCK« der Druck der Diskettentasche vorgeschlagen. An dieser Stelle kann noch entschieden werden, ob die Kennung noch korrigiert werden soll. In diesem Fall kann mit <HOME> der Modus »ERFASSEN« verlassen und in den Korrekturmodus gegangen werden. Nach erfolgter Korrektur kann dann über den Menüpunkt »DRUCK« die Diskettentasche auf dem Drucker ausgegeben werden. Wird »FERTIG« ausgeführt, dann werden die Variablen (siehe Tabelle) normiert und das Programm ist bereit, die nächste Diskette zu erfassen. Sollte beim Erfassen ein Irrtum aufgetreten sein, dann kann ohne Schaden mit <SPACE> zurückgegangen und nochmals erfaßt werden.

Druck:

Im Druckmodus kann wahlweise »TASCHE« zum Druck der Diskettentasche oder »ETIKETTEN« zum Druck der Etiketten angewählt werden.

Druck Tasche:

Eine Diskettentasche für eine zuvor erfaßte Diskette kann beliebig oft ausgegeben werden.

Druck Etiketten:

In diesem Modus gibt es die Punkte »VORLEGEN« und »DRUCK«. Beim Vorlegen wird das nächste zu druckende Etikett vorgeschlagen. Soll nicht gedruckt werden, dann ist wieder »VORLEGEN« anzuwählen. »DRUCK« kann beliebig oft aufgerufen werden. Damit können auch nach dem Papiereinspannen Probedrucke erstellt werden.

Etiketten:

Die Unterpunkte fordern zum »SICHERN« und zum »LADEN« auf. Sollen die erfaßten Disketten erst später gedruckt werden, dann können die Etiketten auf einer freien Diskette gesichert werden. Die Sicherung erfolgt in eine sequentielle Datei mit dem festen Namen »DISK.ETIKETTEN«. Eine bestehende Datei mit dem gleichen Namen wird dabei überschrieben. Soll in einem späteren Lauf die Etikettendatei erweitert werden, dann muß vor dem Erfassen der ersten Diskette über »ETIKETTEN« und »LADEN« die alte Datei geladen werden.

Die gedruckten Diskettentaschen können mit einer Schere oder mit Lineal und Teppichmesser ausgeschnitten werden. Mit dem Teppichmesser können gleich mehrere Diskettentaschen bearbeitet werden. Ausgeschnitten wird entlang der äußeren gepunkteten Begrenzung. Die Klebelaschen neben dem Etikett auf der Diskettentasche müssen stehenbleiben. Nach dem Ausschneiden werden die Klebelaschen nach hinten geknickt. Nachdem das Vorderteil an der gestrichelten Linie nach hinten gefaltet wurde, erhalten die Klebelaschen etwas Alleskleber. (Siegfried Linke/nj)

ET$(300)	Feld für 100 Etiketteninformationen
SA$(32)	Felder für Diskettentaschendruck.
SB$(32)	Seite A/Seite B
AE	Zähler für erfaßte Etiketten
SA + SB	Anzahl Dateis Seite A/Seite B
FA + FB	Freie Blöcke Seite A/Seite B
WW	Drucksteuerung linke/rechte Seite auf der Tasche
F1	Schriftfarbe
F2	Farbe inaktiver Menüpunkt
F3	Farbe aktiver Menüpunkt
C1(4)	Farbsteuerung Menüzeile 1
C2(4)	Farbsteuerung Menüzeile 2
M	Zeiger auf Menüzeile (1 oder 2)
M0	Zeiger innerhalb Menüzeile 1
M1	Zeiger innerhalb Menüzeile 2
MM	Eingabevariable. Wird bei Menü-Steuerung erhöht beziehungsweise erniedrigt. danach wird bei <RETURN> M0/M1 gesetzt.

Tabelle der wichtigsten Variablen

```
10 REM ********************************    <060>
20 REM *                              *    <069>
30 REM *   S.LINKE  &  J.PLESCHKA     *    <203>
40 REM *                              *    <089>
50 REM ********************************    <100>
60 REM                                     <122>
70 REM                                     <132>
80 REM DITA KORS: 1 20.5.86                <114>
90 AZ = 100 * 3 :REM FUER ETIKETTEN        <243>
100 DIM ET$(300)                           <072>
110 AF = 32 :REM ANZAHL FILES AUF TASCHE   <225>
120 DIM SA$(32),SB$(32) :REM FILES SEITE A
    / SEITE B                              <022>
130 FOR I=1 TO AZ:ET$(I)="*":NEXT          <064>
140 AE = 1 :REM ANZAHL ETIKETTEN           <079>
150 ET$(0) = "{3SPACE}(C) S.LINKE & J.PLES
    CHKA{3SPACE}"                          <182>
160 ET$(1) = " TEL.:07533 / 2087{5SPACE}"  <255>
170 ET$(2) = " TEL.:07533 / 3128{5SPACE}"  <167>
180 POKE 53280,0:POKE 53281,0:POKE 646,1   <250>
190 PRINT CHR$(142) :REM GROSS/GRAFIK      <063>
200 POKE 788,52  :REM STOPTASTE SPERREN    <040>
210 PRINT CHR$(8) :REM SHIFT/COMMODORE SPE
    RREN                                   <085>
220 REM --------------------               <061>
230 REM CHAR-CODE FUER ETIKETT AUF VDU     <104>
240 LO = 207:RO = 208                      <154>
250 LU = 204:RU = 186                      <219>
260 LB = 180:RB = 170                      <065>
270 OB = 183:UB = 175                      <031>
280 SI =  18:SO = 146                      <122>
290 REM --------------------               <133>
300 LG = 30                                <054>
310 NA$ = "DISK.ETIKETTEN"                 <175>
320 LE$ = "{4SPACE}"                       <206>
330 F1 = 153 :REM HELLGRUEN                <040>
340 F2 =  28 :REM ROT                      <174>
350 F3 = 158 :REM GELB                     <120>
360 DIM C1(4),C2(4)     :REM FARBEN        <015>
370 DIM M$(24)                             <014>
380 REM --------------------               <223>
390 REM MENUE-ZEILEN                       <131>
400 M$(1)  = "ERFASSEN "                   <224>
410 M$(2)  = "DRUCK{4SPACE}"               <031>
420 M$(3)  = "ETIKETTEN"                   <210>
```

Listing 1. »DITA« druckt Diskettentaschen und Etiketten. Geben Sie das Programm bitte mit dem Checksummer V3.0 ein.

```
430 M$(4)   = "KORREKTUR"                  <130>
440 REM ---------------------              <027>
450 M$(5)   = "SEITE A{2SPACE}"            <090>
460 M$(6)   = "SEITE B{2SPACE}"            <196>
470 M$(7)   = "DRUCK{4SPACE}"              <251>
480 M$(8)   = "FERTIG{3SPACE}"             <009>
490 REM ---------------------              <077>
500 M$(9)   = "TASCHE{3SPACE}"             <160>
510 M$(10)  = "ETIKETTEN"                  <235>
520 M$(11)  = "FERTIG{3SPACE}"             <233>
530 M$(12)  = "{9SPACE}"                   <178>
540 REM ---------------------              <129>
550 M$(13)  = "SICHERN{2SPACE}"            <094>
560 M$(14)  = "LADEN{4SPACE}"              <148>
570 M$(15)  = "FERTIG{3SPACE}"             <028>
580 M$(16)  = "{9SPACE}"                   <229>
590 REM ---------------------              <179>
600 M$(17)  = "KENNUNG{2SPACE}"            <188>
610 M$(18)  = "FERTIG{3SPACE}"             <005>
620 M$(19)  = "{9SPACE}"                   <206>
630 M$(20)  = "{9SPACE}"                   <182>
640 REM ---------------------              <229>
650 M$(21)  = "VORLEGEN "                  <096>
660 M$(22)  = "DRUCK{4SPACE}"              <015>
670 M$(23)  = "FERTIG{3SPACE}"             <031>
680 M$(24)  = "{9SPACE}"                   <233>
690 REM ---------------------              <023>
700 REM DIVERSE  STRINGS FUER DRUCK        <106>
710 HD$ = "{HOME,23DOWN}"                  <057>
720 RI$ = "{5RIGHT}"                       <234>
730 KE$ = " KENNUNG (MAXIMAL 30 ZEICHEN) " <211>
740 Z1$ = KE$                              <010>
750 T1$ = "{16SPACE}"                      <253>
760 T2$ = "{30SPACE}"                      <015>
770 D1$ = "*{2SPACE}"                      <249>
780 D2$ = "{2SPACE}*"                      <011>
790 D3$ = "*******************************
    *****"                                 <038>
800 D4$ = "+......+"                       <078>
810 D6$ = "...............................
    ........................."             <007>
820 DM$ = "{56SPACE}"                      <230>
830 DX$ = "-------------------------------
    ------------------------"              <044>
840 D7$ = "{6SPACE}"                       <073>
850 D8$ = "{25SPACE}"                      <091>
860 DIM PE(10)                             <228>
870 PE$(0) = " "                           <060>
880 PE$(1) = ".{6SPACE}.{10SPACE}"         <073>
890 PE$(2) = "{10SPACE}.{6SPACE}."         <147>
900 PE$(3) = ".{6SPACE}."                  <221>
910 PE$(4) = "{7SPACE}."                   <166>
920 PE$(5) = ".{7SPACE}"                   <240>
930 DL$ = "{30SPACE}"                      <076>
940 GOTO 1580 :REM INS HAUPTPROGRAMM       <185>
950 REM ------------------------------     <220>
960 REM DRUCKER ANMELDEN                   <092>
970 OPEN 3,4                               <047>
980 RETURN                                 <020>
990 REM ------------------------------     <004>
1000 REM ZEICHEN Z DER LAENGE LG AUSGEBEN  <031>
1010 FOR I=1 TO LG:PRINT CHR$(Z);:NEXT     <058>
1020 RETURN                                <060>
1030 REM ------------------------------    <046>
1040 REM RAHMEN AUSGEBEN                   <085>
1050 PRINT CHR$(147); :REM CLR/HOME        <070>
1060 PRINT"{4DOWN}";                       <058>
1070 Z=OB:PRINT LE$;CHR$(LO);:GOSUB 1010:P
     RINT CHR$(RO)                         <217>
1080 Z=32:PRINT LE$;CHR$(LB);:GOSUB 1010:P
     RINT CHR$(RB)                         <249>
1090 Z=UB:PRINT LE$;CHR$(LU);:GOSUB 1010:P
     RINT CHR$(RU)                         <018>
1100 Z=32                                  <203>
1110 FOR J=1 TO 4                          <074>
1120 PRINT LE$;CHR$(LB);:GOSUB 1010:PRINT
     CHR$(RB)                              <059>
1130 NEXT                                  <124>
1140 Z=UB:PRINT LE$;CHR$(LU);:GOSUB 1010:P
     RINT CHR$(RU)                         <068>
1150 RETURN                                <192>
1160 REM ------------------------------    <176>
1170 REM MENUEZEILEN AUSGEBEN              <253>
1180 PRINT CHR$(19);                       <013>
1190 PRINT CHR$(SI);CHR$(C1(1));M$(1);CHR$
     (SO);CHR$(32);                        <020>
1200 PRINT CHR$(SI);CHR$(C1(2));M$(2);CHR$
     (SO);CHR$(32);                        <126>
1210 PRINT CHR$(SI);CHR$(C1(3));M$(3);CHR$
     (SO);CHR$(32);                        <232>
1220 PRINT CHR$(SI);CHR$(C1(4));M$(4);CHR$
     (SO);CHR$(32)                         <122>
1230 P = M0*4                              <203>
1240 REM ------------------------------    <000>
1250 PRINT HD$;                            <040>
1260 PRINT CHR$(SI);CHR$(C2(1));M$(P+1);CH
     R$(SO);CHR$(32);                      <042>
1270 PRINT CHR$(SI);CHR$(C2(2));M$(P+2);CH
     R$(SO);CHR$(32);                      <245>
1280 PRINT CHR$(SI);CHR$(C2(3));M$(P+3);CH
     R$(SO);CHR$(32);                      <193>
1290 PRINT CHR$(SI);CHR$(C2(4));M$(P+4);CH
     R$(SO);CHR$(32);                      <141>
1300 IF Z1$ = "*"THEN Z1$=      DL$        <073>
1310 PRINT LEFT$(HD$,6);RI$;CHR$(F3);CHR$(
     SI);Z1$;CHR$(SO);                     <252>
1320 PRINT LEFT$(HD$,9);RI$;               <155>
1330 IF Z2$ = "*"THEN Z2$=LEFT$(DL$,23)    <159>
1340 IF Z3$ = "*"THEN Z3$=LEFT$(DL$,23)    <201>
1350 PRINT CHR$(SI);CHR$(F3);"A:";CHR$(SO)
     ;CHR$(F1);"{2SPACE}";Z2$              <230>
1360 PRINT"{DOWN}";RI$;                    <154>
1370 PRINT CHR$(SI);CHR$(F3);"B:";CHR$(SO)
     ;CHR$(F1);"{2SPACE}";Z3$              <011>
1380 RETURN                                <168>
1390 REM ------------------------------    <152>
1400 REM HIGHLIGHT LOESCHEN                <081>
1410 FOR I=1 TO 4:C1(I)=F2:C2(I)=F2:NEXT   <245>
1420 GOSUB 1180                            <123>
1430 RETURN                                <218>
1440 REM ------------------------------    <202>
1450 REM AUF ZEICHEN VON TASTATUR WARTEN   <213>
1460 POKE 198,0                            <098>
1470 GET X$:IF X$="" THEN 1470             <065>
1480 Z=ASC(X$)                             <146>
1490 RETURN                                <022>
1500 REM ------------------------------    <006>
1510 REM ABBRUCH NACH DISKERROR            <028>
1520 CLOSE 2:CLOSE 15                      <170>
1530 FOR I=1 TO 5000:NEXT                  <177>
1540 PRINT LEFT$(HD$,3);T2$;               <170>
1550 GOTO 1610                             <094>
1560 REM ------------------------------    <068>
1570 REM MENUESTEUERUNG                    <005>
1580 FA=664:FB=664                         <141>
1590 Z2$="*":Z3$="*"                       <096>
1600 PRINT CHR$(F1);:GOSUB 1050            <061>
1610 M = 1:MM=1:M0=1:M1 = 0 :REM NORM MENU
     ESTEUERUNG                            <152>
1620 FOR I=1 TO 4:C1(I)=F2:C2(I)=F2:NEXT   <201>
1630 C1(M0) = F3                           <213>
1640 C2(M1) = F3                           <039>
1650 GOSUB 1180 :REM MENUEZEILEN AUSGEBEN  <082>
1660 GOSUB 1460 :REM ZEICHEN VON TASTATUR  <025>
1670 REM WENN F8 DANN STOP UND SHIFT/COMMO
     DORE ZULASSEN UND SCHLUSS             <167>
1680 IF Z=140  THEN POKE 788,49:PRINT CHR$
     (9):END                               <137>
1690 IF Z=13  THEN 1800                    <120>
1700 IF Z=19  THEN 1610                    <004>
1710 IF Z<>32 THEN 1660                    <165>
1720 MM=MM+1                               <016>
1730 IF M  > 2 THEN M  = 1                 <106>
1740 IF MM > 4 THEN MM = 1                 <148>
1750 IF M=1 THEN M0=MM:GOTO 1620           <091>
1760 M1=MM                                 <209>
1770 IF M$((M0*4)+M1) = "{9SPACE}" THEN  M
     1=1:MM=1                              <004>
1780 GOTO  1620                            <100>
1790 REM ------------------------------    <042>
1800 GOSUB 1410 :REM HIGHLIGHT LOESCHEN    <021>
1810 IF M = 1 THEN M=2:M1=1:MM=1:GOTO 1750 <092>
1820 IF (M0=1) THEN 2400 :REM ERFASSEN     <025>
1830 IF (M0=2) THEN 3760 :REM DRUCK        <008>
1840 IF (M0=3) THEN 4580 :REM ETIKETTEN    <234>
1850 IF (M0=4) THEN 1880 :REM KORREKTUR    <196>
1860 REM ------------------------------    <114>
1870 REM **** KORREKTUR *****              <060>
1880 IF M1=2 THEN 1610 :REM FERTIG         <200>
1890 K$ = "":ZZ=0                          <043>
1900 PRINT LEFT$(HD$,6);">";               <247>
1910 GOSUB 1460 :REM ZEICHEN VON TASTATUR  <021>
1920 IF Z=13 THEN PRINT LEFT$(HD$,6);" ";:
     MM=MM+1:GOTO 1730                     <042>
1930 IF Z=19 THEN ZZ=0:K$="":GOTO 1990 :RE
     M CLR                                 <248>
```

```
1940 IF Z<>20 THEN GOSUB 2190:GOTO 1980     <185>
1950 IF ZZ<=1 THEN ZZ=0:K$="":GOTO 1990     <185>
1960 ZZ=ZZ-1:K$=LEFT$(K$,ZZ) :GOTO 1990     <173>
1970 IF ZZ=30 THEN 2010                     <246>
1980 K$ = K$+X$:ZZ=ZZ+1                     <046>
1990 KE$=LEFT$(K$+T2$,30)                   <181>
2000 ET$((AE*3)    ) = KE$                  <185>
2010 Z1$ = KE$                              <010>
2020 GOSUB 1310                             <022>
2030 GOTO  1910                             <112>
2040 REM --------------------------------   <038>
2050 REM INITIALISIERUNG DISK               <071>
2060 OPEN 15,8,15,"I0"                      <157>
2070 REM FEHLERABFRAGE DISK                 <244>
2080 INPUT#15,A$,B$,C$,D$                   <197>
2090 IF A$="00" THEN RETURN                 <048>
2100 PRINT LEFT$(HD$,3);T2$;                <222>
2110 PRINT LEFT$(HD$,3);A$;",";B$;",";C$;"
     ,";D$                                  <015>
2120 RETURN                                 <146>
2130 REM --------------------------------   <130>
2140 REM ZEICHEN VON DISK KOLEN             <165>
2150 REM AUF DRUCKBARKEIT PRUEFEN           <067>
2160 REM ******************************     <140>
2170 GET#2,X$:IF X$="" THEN X$=CHR$(0)      <016>
2180 Z=ASC(X$)                              <084>
2190 IF Z =160 THEN X$=".":RETURN           <042>
2200 IF Z < 32 OR Z > 127 THEN X$="?" :REM
     UNZUL.ZEICHEN                          <210>
2210 RETURN                                 <236>
2220 REM --------------------------------   <220>
2230 REM STARTADRESSE ERMITTELN             <113>
2240 AA$ = "{6SPACE}"                       <016>
2250 IF FT <> 2 THEN RETURN :REM NICHT PRG
     -FILE                                  <113>
2260 IF SP=0 AND SE=0 THEN RETURN           <093>
2270 OPEN 3,8,3,"#"                         <127>
2280 PRINT#15,"B-R";3;0;SP;SE               <166>
2290 GOSUB 2080 :REM DISKERROR  ABFRAGEN    <119>
2300 IF A$<>"00" THEN 2370                  <060>
2310 PRINT#15,"B-P";3;2                     <216>
2320 GET#3,X$:IF X$=""THEN X$=CHR$(0)       <184>
2330 AL=ASC(X$)                             <066>
2340 GET#3,X$:IF X$=""THEN X$=CHR$(0)       <204>
2350 AH=ASC(X$)*256                         <146>
2360 AA=AL+AH:AA$=RIGHT$("{6SPACE}"+STR$(A
     A),6)                                  <043>
2370 CLOSE 3:RETURN                         <227>
2380 REM --------------------------------   <126>
2390 REM ERFASSEN                           <204>
2400 IF M1=4 THEN AE=AE+1:SA=0:SB=0:GOTO 1
     580 :REM FERTIG                        <241>
2410 IF M1=3 THEN M=2:M0=2:M1=1:MM=1:GOTO
     1620                                   <075>
2420 GOSUB 2060                             <034>
2430 IF A$ <> "00" THEN 1520                <175>
2440 OPEN 2,8,2,"#"   :REM DATENKANAL       <172>
2450 PRINT#15,"B-R";2;0;18;0                <131>
2460 PRINT#15,"B-P";2;144                   <108>
2470 DN$=""                                 <108>
2480 FOR I=1 TO 16:GOSUB 2170:DN$=DN$+X$:N
     EXT                                    <195>
2490 GET#2,X$:GET#2,X$                      <171>
2500 ID$ =""                                <078>
2510 GOSUB 2170:ID$=ID$+X$                  <217>
2520 GOSUB 2170:ID$=ID$+X$                  <227>
2530 XX$=DN$+" ID= "+ID$                    <091>
2540 ET$((AE*3)+M1) = XX$                   <112>
2550 ET$(AE*3) = Z1$                        <023>
2560 Z2$=ET$((AE*3)+1):Z3$=ET$((AE*3)+2):G
     OSUB 1310                              <110>
2570 PRINT LEFT$(HD$,14);                   <210>
2580 FOR I=1 TO 8                           <014>
2590 PRINT T2$;                             <175>
2600 PRINT LEFT$(T2$,6)                     <206>
2610 NEXT                                   <080>
2620 PRINT LEFT$(HD$,13)                    <077>
2630 PRINT CHR$(SI);                        <011>
2640 IF M1=1 THEN PRINT "A:";               <192>
2650 IF M1=2 THEN PRINT "B:";               <014>
2660 PRINT CHR$(SO);"{2SPACE}";             <079>
2670 T=18:S=1                               <205>
2680 Y2=0 :REM ZAEHLER DER FILES            <233>
2690 Y3=14:REM FUER FILEUEBERNAHME          <236>
2700 Y4=0 :REM FUER ANFRAGE                 <062>
2710 IF M1=1 THEN SA=0:FOR I=0 TO AF:SA$(I
     )= "*":NEXT                            <008>
2720 IF M1=2 THEN SB=0:FOR I=0 TO AF:SB$(I
     )= "*":NEXT                            <148>
2730 PRINT LEFT$(HD$,22);CHR$(SI);"RETURN"
     ;CHR$(SO);" = UEBERNEHMEN ";           <101>
2740 PRINT CHR$(SI);"SPACE";CHR$(SO);" = I
     GNORIEREN"                             <028>
2750 REM --------------------------------   <242>
2760 REM FREIE BLOECKE ERMITTELN            <184>
2770 BF = 0                                 <082>
2780 FOR I=4 TO 140 STEP 4                  <141>
2790 IF I=72 THEN 2830 :REM SPUR 18         <223>
2800 PRINT#15,"B-P";2;I                     <122>
2810 GET#2,X$:IF X$=""THEN X$=CHR$(0)       <148>
2820 BF = ASC(X$)+BF                        <036>
2830 NEXT                                   <046>
2840 IF M1=1 THEN FA=BF                     <086>
2850 IF M1=2 THEN FB=BF                     <162>
2860 REM --------------------------------   <098>
2870 PRINT#15,"U1";2;0;T;S                  <197>
2880 GOSUB 2080:REM DISKFEHLER ABFRAGEN     <106>
2890 IF A$ <> "00" THEN 1520                <127>
2900 PRINT#15,"B-P";2;0                     <154>
2910 GET#2,X$:IF X$=""THEN X$=CHR$(0)       <250>
2920 T=ASC(X$)                              <038>
2930 GET#2,X$:IF X$=""THEN X$=CHR$(0)       <014>
2940 S=ASC(X$)                              <054>
2950 REM SCHLEIFE FUER 8 FILES ANFANG       <111>
2960 FOR X=0 TO 7                           <099>
2970 IF Y2>AF-1 THEN 3500 :REM ZUVIELE FIL
     ES                                     <221>
2980 PRINT#15,"B-P";2;X*32+2                <028>
2990 GET#2,X$:IF X$=""THEN X$=CHR$(0):REM
     FILETYP                                <101>
3000 TY = ASC(X$)                           <152>
3010 GET#2,X$:IF X$=""THEN X$=CHR$(0):REM
     SPUR                                   <192>
3020 SP = ASC(X$)                           <096>
3030 GET#2,X$:IF X$=""THEN X$=CHR$(0):REM
     SEKTOR                                 <245>
3040 SE = ASC(X$)                           <028>
3050 PRINT#15,"B-P";2;X*32+30               <109>
3060 GET#2,X$:IF X$=""THEN X$=CHR$(0):REM
     FILETYP                                <171>
3070 NB = ASC(X$)                           <014>
3080 GET#2,X$:IF X$=""THEN X$=CHR$(0):REM
     FILETYP                                <193>
3090 HB = ASC(X$)*256                       <104>
3100 BL=HB+NB                               <138>
3110 FT = TY  AND 15      :REM FILETYP      <099>
3120 IF FT = 0 THEN 3500 :REM DEL-FILE NIC
     HT AUSGEBEN                            <088>
3130 PRINT#15,"B-P";2;X*32+5                <204>
3140 FF$=""                                 <216>
3150 FOR II=1 TO 16                         <219>
3160 GOSUB 2170 :REM FILENAME HOLEN         <159>
3170 FF$=FF$+X$                             <012>
3180 NEXT II                                <108>
3190 FF$=FF$+" "                            <055>
3200 IF FT=1 THEN FT$="SEQ"                 <108>
3210 IF FT=2 THEN FT$="PRG"                 <148>
3220 IF FT=3 THEN FT$="USR"                 <068>
3230 IF FT=4 THEN FT$="REL"                 <169>
3240 IF FT>4 THEN FT$="???"                 <039>
3250 K$  = ">"                              <039>
3260 IF TY AND  64 THEN 3300                <050>
3270 K$  = " "                              <035>
3280 IF TY AND 128 THEN 3300                <156>
3290 K$  = "*"                              <074>
3300 FT$= K$+FT$                            <219>
3310 BL$=RIGHT$("{3SPACE}"+STR$(BL),3)      <108>
3320 BL$=BL$+"{2SPACE}"                     <219>
3330 GOSUB 2240 :REM STARTADRESSE ERMITTEL
     N                                      <088>
3340 XX$=BL$+FF$+FT$+AA$                    <119>
3350 IF Y4 THEN 3440                        <217>
3360 PRINT LEFT$(HD$,Y3);LEFT$(RI$,4);      <028>
3370 PRINT CHR$(SI);XX$;CHR$(SO);           <018>
3380 GOSUB 1460 :REM ZEICHEN VON TASTATUR   <223>
3390 IF Z=136 THEN Y4=1:GOTO 3410           <249>
3400 IF Z<>32 AND Z<>13 THEN 3380           <212>
3410 PRINT LEFT$(HD$,Y3);LEFT$(RI$,4);      <147>
3420 PRINT T2$;"{2SPACE}"                   <038>
3430 IF Z=32 THEN 3500 :REM IGNORIEREN      <108>
3440 PRINT LEFT$(HD$,Y3);LEFT$(RI$,4);      <177>
3450 PRINT XX$                              <197>
3460 IF M1=1 THEN SA$(SA)=XX$:SA=SA+1       <217>
3470 IF M1=2 THEN SB$(SB)=XX$:SB=SB+1       <102>
3480 Y2=Y2+1:Y3=Y3+1                        <042>
3490 IF Y3 >= 22 THEN Y3=14                 <191>
```

Listing 1. »DITA« (Fortsetzung)

```
3500 NEXT X                                     <146>
3510 REM SCHLEIFE FUER 8 FILES (ENDE)           <186>
3520 IF Y2>AF-1 THEN T=0:REM DAS SIND GENU
     G                                          <159>
3530 IF T<>0 THEN GOTO 2870                     <141>
3540 CLOSE 2                                    <001>
3550 CLOSE 15                                   <086>
3560 MM=MM+1:GOTO 1740                          <118>
3570 REM -------------------------------        <044>
3580 REM LEERZEILE AUSGEBEN                     <042>
3590 PRINT#3,PE$(PA);D1$;DL$;D2$;PE$(PE)        <107>
3600 RETURN                                     <102>
3610 REM -------------------------------        <086>
3620 REM ETIKETT AUSGEBEN                       <165>
3630 PRINT#3,PE$(PA);D3$;PE$(PE)                <037>
3640 IF Z1$ = "*" THEN GOSUB 3590:GOTO 366
     0                                          <146>
3650 PRINT#3,PE$(PA);D1$;Z1$;D2$;PE$(PE)        <102>
3660 PRINT#3,PE$(PA);D3$;PE$(PE)                <067>
3670 PRINT#3,PE$(PA);D1$;DL$;D2$;PE$(PE)        <187>
3680 IF Z2$ = "*" THEN GOSUB 3590:GOTO 370
     0                                          <114>
3690 PRINT#3,PE$(PA);D1$;" A: ";Z2$;"{3SPA
     CE}";D2$;PE$(PE)                           <122>
3700 IF Z3$ = "*" THEN GOSUB 3590:GOTO 372
     0                                          <182>
3710 PRINT#3,PE$(PA);D1$;" B: ";Z3$;"{3SPA
     CE}";D2$;PE$(PE)                           <018>
3720 PRINT#3,PE$(PA);D1$;DL$;D2$;PE$(PE)        <237>
3730 PRINT#3,PE$(PA);D3$;PE$(PE)                <137>
3740 RETURN                                     <242>
3750 REM -------------------------------        <226>
3760 IF M1 = 3 THEN SA=0:SB=0:AE=AE+1:GOTO
      1580 :REM FERTIG                          <038>
3770 REM DRUCK                                  <122>
3780 IF M1 = 1 THEN 4320 :REM TASCHEN           <052>
3790 REM ETIKETTEN VORLEGEN/DRUCKEN             <049>
3800 PP = 0:REM ANFANG ETIKETTEN                <154>
3810 GOSUB 970 :REM DRUCKER ANMELDEN            <144>
3820 P=20 :REM STEUERUNG MENUE                  <149>
3830 M1=1                                       <033>
3840 FOR I=1 TO 4:C2(I)=F2:NEXT                 <003>
3850 C2(M1) = F3                                <219>
3860 GOSUB 1250 :REM MENUEZEILE AUSGEBEN        <255>
3870 GOSUB 1460 :REM ZEICHEN VON TASTATUR       <205>
3880 IF Z=13 THEN 3930                          <040>
3890 IF Z<>32 THEN 3870                         <079>
3900 M1=M1+1                                    <190>
3910 IF M1>3 THEN M1=1                          <230>
3920 GOTO 3840                                  <066>
3930 IF M1=3 THEN CLOSE 3:GOTO 1580             <049>
3940 IF M1=2 THEN PA=0:PE=0:GOSUB 3630:PRI
     NT#3:M1=1:GOTO 3840                        <080>
3950 PP = PP+1                                  <030>
3960 Z1$ = ET$((PP*3)-3)                        <224>
3970 Z2$ = ET$((PP*3)-2)                        <240>
3980 Z3$ = ET$((PP*3)-1)                        <000>
3990 GOSUB 1310                                 <216>
4000 M1=M1+1                                    <034>
4010 GOTO 3840                                  <156>
4020 REM -------------------------------        <242>
4030 REM LEERZEILE TASCHE                       <031>
4040 PRINT#3,PE$(PA);DM$;PE$(PE)                <072>
4050 ZZ=ZZ+1                                    <189>
4060 RETURN                                     <052>
4070 REM -------------------------------        <036>
4080 REM TRENNER TASCHE VORN/HINTEN             <068>
4090 GOSUB 4040:GOSUB 4040                      <250>
4100 PRINT#3,D4$;DX$;D4$                        <039>
4110 PA=4:PE=5                                  <094>
4120 GOSUB 4040:GOSUB 4040                      <026>
4130 RETURN                                     <124>
4140 REM -------------------------------        <108>
4150 REM ZEILE FUER TASCHE SEITE A              <105>
4160 IF ZZ = 10  THEN GOSUB 4090                <107>
4170 IF SA$(I) = "*" THEN SA$(I) = D8$          <030>
4180 IF WW=0 THEN WW=1:XX$=MID$(SA$(I),6,1
     9)+RIGHT$(SA$(I),6):RETURN                 <032>
4190 XX$=XX$+"{2SPACE}"+MID$(SA$(I),6,19)+
     RIGHT$(SA$(I),6)                           <213>
4200 PRINT#3,PE$(PA);"{2SPACE}";XX$;"{2SPA
     CE}";PE$(PE)                               <217>
4210 ZZ=ZZ+1:WW=0                               <027>
4220 RETURN                                     <214>
4230 REM -------------------------------        <198>
4240 REM ZEILE FUER TASCHE SEITE B              <068>
4250 IF ZZ = 10  THEN GOSUB 4090                <197>
4260 IF SB$(I) = "*" THEN SB$(I) = D8$          <009>
4270 IF WW=0 THEN WW=1:XX$=MID$(SB$(I),6,1
     9)+RIGHT$(SB$(I),6):RETURN                 <146>
4280 XX$=XX$+"{2SPACE}"+MID$(SB$(I),6,19)+
     RIGHT$(SB$(I),6)                           <177>
4290 GOTO 4200                                  <222>
4300 REM -------------------------------        <012>
4310 REM TASCHE DRUCKEN                         <029>
4320 GOSUB 970 :REM DRUCKER ANMELDEN            <146>
4330 PA=3:PE=3                                  <018>
4340 PRINT#3,D4$;D6$;D4$                        <142>
4350 GOSUB 4040                                 <138>
4360 PA=1:PE=2 :REM FUER ETIKETT                <136>
4370 GOSUB 3630                                 <216>
4380 PA=3:PE=3                                  <070>
4390 GOSUB 4040                                 <180>
4400 ZZ =  0                                    <190>
4410 IF SA=0 THEN FOR I=0 TO AF:SA$(I)="*"
     :NEXT                                      <113>
4420 IF SB=0 THEN FOR I=0 TO AF:SB$(I)="*"
     :NEXT                                      <141>
4430 XX$=RIGHT$("{4SPACE}"+STR$(FA),3)          <075>
4440 PRINT#3,PE$(PA);" A={2SPACE}";XX$;" B
     LOECKE FREI{4SPACE}";D8$;D7$;PE$(PE)       <103>
4450 WW=0                                       <204>
4460 FOR I=0 TO AF-1:GOSUB 4160:NEXT            <125>
4470 XX$=RIGHT$("{4SPACE}"+STR$(FB),3)          <179>
4480 GOSUB 4040                                 <014>
4490 PRINT#3,PE$(PA);" B={2SPACE}";XX$;" B
     LOECKE FREI{4SPACE}";D8$;D7$;PE$(PE)       <217>
4500 WW=0                                       <254>
4510 FOR I=0 TO AF-1:GOSUB 4250:NEXT            <110>
4520 GOSUB 4040:GOSUB 4040                      <172>
4530 PRINT#3,PE$(PA);D6$;PE$(PE)                <112>
4540 FOR I=1 TO 18:PRINT#3:NEXT :REM NEUE
     SEITE                                      <036>
4550 CLOSE 3:MM=MM+2:GOTO 1730                  <107>
4560 REM -------------------------------        <018>
4570 REM ETIKETTEN LADEN                        <182>
4580 IF M1=3 THEN 1610 :REM FERTIG              <168>
4590 GOSUB 2060 :REM DISK INITIALISIEREN        <007>
4600 IF A$ <> "00" THEN 1520 :REM DISK-ERR
     OR                                         <061>
4610 IF M1=1 THEN 4840 :REM SICHERN             <165>
4620 OPEN 2,8,2,NA$+",S,R"                      <020>
4630 GOSUB 2080                                 <023>
4640 IF A$ <> "00" THEN 1520 :REM FILE NIC
     HT DA                                      <122>
4650 AE=0                                       <174>
4660 GOSUB 4780                                 <181>
4670 ET$(AE*3) = XX$                            <054>
4680 GOSUB 4780                                 <201>
4690 ET$((AE*3)+1) = XX$                        <220>
4700 GOSUB 4780                                 <221>
4710 ET$((AE*3)+2) = XX$                        <016>
4720 IF ST = 64 THEN 4750                       <053>
4730 IF ET$(AE*3) = "*" THEN 4750               <189>
4740 AE=AE+1:GOTO 4660                          <113>
4750 CLOSE 2:CLOSE 15                           <098>
4760 MM=3:GOTO 1740                             <046>
4770 REM -------------------------------        <230>
4780 XX$=""                                     <036>
4790 GET#2,X$                                   <073>
4800 IF ASC(X$) = 13 THEN RETURN                <119>
4810 XX$=XX$+X$:GOTO 4790                       <251>
4820 REM -------------------------------        <024>
4830 REM ETIKETTEN SICHERN                      <067>
4840 OPEN 2,8,2,NA$+",S,W"                      <250>
4850 GOSUB 2080                                 <243>
4860 IF A$  = "00" THEN 4900 :REM OK            <095>
4870 CLOSE 2                                    <063>
4880 PRINT#15,"S:"+NA$        :REM FILE-LOE
     SCHEN                                      <188>
4890 GOTO 4840                                  <028>
4900 FOR I=0 TO AE+1                            <242>
4910 XX$=ET$(3*I)                               <232>
4920 GOSUB 5020                                 <146>
4930 XX$=ET$((3*I)+1)                           <096>
4940 GOSUB 5020                                 <166>
4950 XX$=ET$((3*I)+2)                           <117>
4960 GOSUB 5020                                 <186>
4970 NEXT                                       <154>
4980 CLOSE 2:CLOSE 15                           <074>
4990 PRINT"{HOME,2DOWN}";T2$;                   <066>
5000 MM=3:GOTO 1740                             <032>
5010 REM -------------------------------        <216>
5020 FOR J=1 TO LEN(XX$):X$=MID$(XX$,J,1):
     PRINT#2,X$;:NEXT                           <081>
5030 PRINT#2,CHR$(13);                          <140>
5040 RETURN                                     <016>
5050 REM -------------------------------        <000>
```

Listing 1. »DITA« (Schluß)

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael Scharfenberger

Stellv. Chefredakteur: Albert Absmeier

Leitender Redakteur: Georg Klinge

Redaktion: Herbert Buckel (bj), Achim Hübner (ah), Norbert Jungmann (nj), Dieter Mayer (dm), Harald Meyer (hm), Markus Ohnesorg (og), Thomas Röder (tr), Boris Schneider (bs), Karsten Schramm (ks)

Titelfoto: Jens Jancke

Titelgestaltung: Heinz Rauner Grafik-Design

Layout:
Leo Eder (Ltg.), Sigrid Kowalewski (Cheflayouterin), Rolf Raß, Katja Milles

Produktionsleiter: Klaus Buck

Anzeigenverkaufsleitung: Ralph-Peter Rauchfuss

Auslandsrepräsentation:

Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 56 56, Telex: 862 329

USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063 Telefon: (4 15) 366-3600

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Marketingleiter: Hans Hörl (114)

Vertriebsleiter: Helmut Grünfeldt (189)

Anzeigenverwaltung und Disposition: Michaela Hörl

Verlagsleiter M&T-Buchverlag: Günther Frank (212)

Druck: SOV St. Otto-Verlag GmbH, Laubanger 23, 8600 Bamberg

Bezugsmöglichkeiten: Leser-Service: Telefon (0 89) 46 13-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 14,–

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (07 11) 64 83-0

Verantwortlich:
Für redaktionellen Teil: Michael Scharfenberger
Für Anzeigen: Britta Fiebig

Redaktionsdirektor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (0 89) 46 13-0, Telex 5-22 052